Innovation in the Software Sector

by Douglas Lippoldt
and Piotr Stryszowski

OECD Innovation Strategy

OECD

Innovation in the Software Sector

Douglas Lippoldt and Piotr Stryszowski

OECD

ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT

The OECD is a unique forum where the governments of 30 democracies work together to address the economic, social and environmental challenges of globalisation. The OECD is also at the forefront of efforts to understand and to help governments respond to new developments and concerns, such as corporate governance, the information economy and the challenges of an ageing population. The Organisation provides a setting where governments can compare policy experiences, seek answers to common problems, identify good practice and work to co-ordinate domestic and international policies.

The OECD member countries are: Australia, Austria, Belgium, Canada, the Czech Republic, Denmark, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Japan, Korea, Luxembourg, Mexico, the Netherlands, New Zealand, Norway, Poland, Portugal, the Slovak Republic, Spain, Sweden, Switzerland, Turkey, the United Kingdom and the United States. The Commission of the European Communities takes part in the work of the OECD.

OECD Publishing disseminates widely the results of the Organisation's statistics gathering and research on economic, social and environmental issues, as well as the conventions, guidelines and standards agreed by its members.

This work is published on the responsibility of the Secretary-General of the OECD. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the Organisation or of the governments of its member countries.

ISBN 978-92-64-06361-7 (print)
ISBN 978-92-64-07676-1 (PDF)

Also available in French: L'*innovation dans le secteur des logiciels*

Cover illustration: © Veer/Fancy Photography.

Corrigenda to OECD publications may be found on line at: *www.oecd.org/publishing/corrigenda*.

Foreword

OECD governments recognise the increasing importance of software for their economies and the potential for public policy to have an impact on innovation in the sector. However, as the software sector becomes increasingly complex and dynamic, more quantitative and empirical analysis of how innovation is spurred is required to better understand the sector. Consequently, the OECD Committee on Industry, Innovation and Entrepreneurship (CIIE) tasked the Secretariat to explore issues such as economic processes, R&D and market environment, and user perspectives of software functionalities.

This project was launched in March 2007 with the establishment of an advisory expert group. Major conferences took place in Caceres, Spain (in November 2007, co-organised by the OECD, the Spanish Ministry of Industry, Tourism and Trade and the regional government of Extremadura) and Tokyo (in October 2008, co-organised by the OECD, the Japanese Ministry of Economy, Trade and Industry and Research Institute of Economy, Trade and Industry). The expert group was also hosted by the Israeli Ministry of Industry, Trade and Labour in June 2008. The authors are grateful to the support received from Israel, Japan and Spain, as well as those countries and industries that nominated experts to the advisory group. They made an essential contribution to the project's success.

This report was prepared by the Structural Policy Division of the OECD Directorate for Science, Technology and Industry. Overall direction was provided by Marcos Bonturi. The main authors were Douglas Lippoldt and Piotr Stryszowski. Desirée van Welsum and Jeoung Yeol Yu provided very significant contributions. Other important contributions were made by Sarah Cox, Kaoru Endo, Linda Haie-Fayle, Sami Mahroum, Elena Navascués and TengTeng Xu.

Table of Contents

Executive Summary

Architecturally, software types can generally be described as belonging to one of three basic categories: *i*) applications, *ii*) operating systems, or *iii*) middleware. *Applications* are software programs providing functionality focused on a particular task for end users, and as noted above, may reside locally on a machine or be delivered remotely. Examples of common software applications for individual household and office users include text processing programs, electronic spreadsheets, desktop publishing programs or various e-mail and web browsing applications. *Operating systems* reside locally, and serve as the platform interface between the hardware system and other types of user software programs while also providing services (*e.g.* file print). *Middleware* is a broad category of software ranging from task-specific functionality to platform-type functionality that permits applications to operate across operating systems and interoperate despite being written in different computer languages. Most software products have a fairly limited product life cycle; it is not uncommon for new versions of many software applications and operating systems to be released every few years. This is in marked contrast to other traditionally manufactured goods, such as furniture, major appliances and automobiles, which often can be expected to last for a decade or more.

A particular category of software is *embedded software,* which denotes a type of software permanently embedded in a given hardware unit. This software is integral to the product, where it resides on a long-term basis, and by necessity it must be extremely reliable. Unlike other types of software, embedded software is sold as part of the hardware product on which it resides. Embedded software becomes ubiquitous in modern economies, as it is used in a very broad range of electronic products and systems. Medical device manufacturers, the consumer electronics industry, the automobile industry, the mobile phone industry, robotics makers and the telecommunications industry all make heavy use of embedded software applications. The market for embedded software is growing at a rapid pace and a large part of it is considered to be outside the traditional software-producing sector.

Increasingly high-speed digital networks are transforming software delivery from a physical carrying device used to install the software on a machine to remote software delivery via the Internet. Out of this a concept called *software-as-a-service* (SaaS) is evolving, which while in an early stage of market adoption, shows many signs of gaining market share in the coming years. SaaS has the advantage of being able to deliver software applications without customers needing to incur some of the traditional overhead expenses involved with application licenses, servers, and other on-site resources. There are several differences between traditional software-as-an-application and SaaS. In the case of the former, users usually pay upfront for the license, a dedicated instance of software is installed on their hardware and they are responsible for deployment, operation maintenance and upgrades of the application. SaaS is in most cases a “pay as you go” system; software is managed and maintained by the SaaS provider, who is responsible for the infrastructure and upgrades.

SaaS is part of a larger general concept called *cloud computing* that refers to the application, platform, or utility services accessed by users over the Internet (a cloud depicts the Internet on computer network diagrams). Cloud-based services are still developing but many anticipate that they will ultimately enable flexible, cost-effective, massively scalable, readily accessible, and highly reliable functionalities that offer users a way to add computing capacity or capabilities without a considerable investment in new computer infrastructure. Business models are developing rapidly and one can expect users to get software-generated functionality from a combination of locally resident software and functionality provided over the web.

Measurement challenges

Identifying the boundaries of the software industry is a continuing challenge

The growing ubiquity of software in today's economy means that it is actually difficult to define precisely the "software industry", as this category fails to encompass all the contributors that taken together develop software for today's diverse market. Essentially, just as software is a pervasive component of many everyday goods and services, software development is increasingly taking place across the economy. Many firms beyond the traditional software sector have embraced software development as a complement to their non-software products. As discussed in several studies (*e.g.* Parker and Grimm, 2000; Grimm *et al.*, 2002), a substantial share of all software produced is not developed by software companies for the general market, but is custom software created by specific users to fulfil their own needs.

Numerous companies that primarily specialise in producing software for the commercial market embody what is considered to be the traditional software industry; this now also encompasses those parts of other industries and traditional market players that are involved in software development, including electronics, finance and government, as well as some academic institutions and non-profit institutions, as part of the traditional sector. But the growing pace of Internet development means that individuals can also act as software developers, as well as firms whose products are being transformed by software technologies, such as automobile producers and medical equipment manufacturers. Thus the term "software sector" more accurately captures the complexity of stakeholders and businesses that participate in developing software for today's economy.

There is no clear and universal measure of software

Clear and reliable statistics are an important tool for assessing innovation and economic performance; however, in the case of software innovation, measurement is not straightforward. Software is an intangible product with very low reproduction costs, which implies that marginal costs of production may serve poorly as a starting point for an exercise aimed at determining the economic value of an additional unit of software. Furthermore, a substantial part of the revenue from software production and distribution arises from other factors, such as advertisements provided together with a software product or from the sale of related services. Because software is sometimes provided free of charge (at least direct charge), it poses a measurement and valuation challenge to statisticians. This creates another difficulty in the process of software quantification, as there may be no monetary payment (*e.g.* wages or price) tracked separately for use in directly valuing the software.

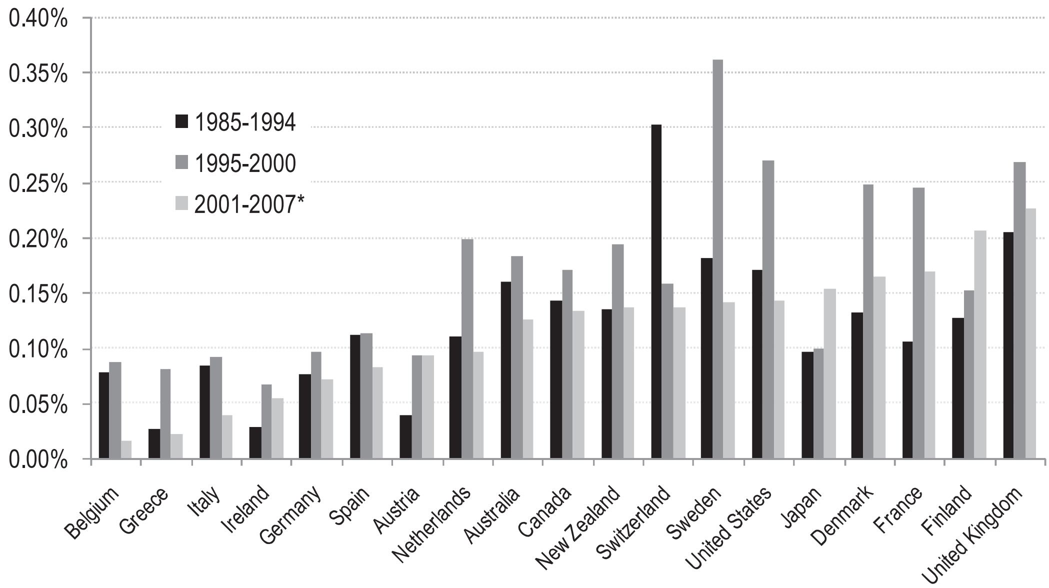
As a result, there are no data or even reliable proxies for some types of software products. Even for types of software where measurement appears more straightforward *a priori*, data collection remains fraught with methodological challenges, such as variations in the accounting treatment of software investment, differences in the valuation techniques of in-house software, and the international comparability of price levels (Ahmad, 2003). In sum, it is difficult to determine accurately the economic value of software.

Existing measures

The software sector has effects that spill over beyond its specific niche, particularly as a widening array of economic activity, goods, and services rely to some extent on software-related technologies. Since these technologies promise to command a greater share of economic activity, the size and effectiveness of investment in software-related R&D may determine economic performance and international competitiveness more broadly. Figure 0.1 shows the contributions of software investment to the GDP growth of different countries. Software's contribution to GDP growth was significantly positive in all of the countries and time periods for which data are available.

Figure 0.1. Contribution of software investment to GDP growth

Average annual growth rates



* or latest available year.

Source: OECD Productivity Database.

Market penetration rates, based on unit counts from surveys rather than valuation estimates, constitute a potentially useful statistical resource. While not an indicator of innovation, they do provide information on the diffusion of software products. From this indicator, for example, one can see some confirmation that the distribution of some types of software via non-traditional channels has increased and in some cases attained significant market shares. However, these results only relate to specific types of software and may also suffer from measurement difficulties and biases. For example, survey respondents often do not account for multiple software products being used by a single customer.

Finally, the fact that software development is particularly intense in human capital suggests that indicators that measure software-related employment could be a useful proxy for software dynamism and importance. Indeed, ICTs, including software, have recently become a major source of employment creation across the OECD countries as well as in a number of leading developing countries. Moreover, in view of the nature of software as an intangible product based on intellectual processes, the sector is particularly dependent on a highly skilled labour force. Employment in the ICT sector has experienced growth in most OECD countries for the last decade.

Software innovation processes

Innovation may be broadly defined as the successful commercial introduction of a new product, service or process. More specifically, according to the OECD's *Oslo Manual* (OECD, 2005), innovation refers to the implementation of "technologically new products and processes and significant technological improvements in products and processes". An innovation has been implemented if it has been introduced on the market (product or process innovation) or used within a production process (process innovation). While technological innovation involves a series of scientific, technological, organisational, financial and commercial activities, it is important in this context to note that research and development (R&D) is just one element – albeit an important one – in the process of software innovation.

Software innovation can be seen as a process leading to:

- development of a novel aspect, feature or application of an existing software product or process; or
- introduction of a new software product, service or process or an improvement in the previous generation of the software product or process; and
- entry to an existing market or the creation of a new market.

Different breeds of actors

Today the software sector is characterised by participants ranging from firms primarily involved in creating software, firms whose products and services are being transformed by software technology, and individual developers working on their own or in some collaborative capacity with a larger firm or organisation. Today, many firms beyond the traditional software sector have embraced software development as a complement to their non-software products; increased emphasis on integration of software is fuelling technological convergence between hardware, software and telecommunication technologies. Technology synergies and interdependencies across different segments are increasing and reinforcing R&D collaboration. Many non-software sector firms are engaged in some form of collaborative R&D initiatives and find that skill, experience, and cost are considered to be vital elements for the success of R&D collaboration. Beyond direct profits from any eventual product, benefits include mutual development in the area of human capital, access to intellectual property, and organisational support, among others. These tight links between the software sector and other industries are highlighted in editions of the *OECD Information Technology Outlook*.

As a result of this environment, a proliferation of business models exists which both influences and is influenced by the process that characterises software development and innovation.

Drivers of software innovation

Software innovation is often driven by user needs and expectations, and at times in the development process, software designers often solicit user feedback. Yet software development primarily takes place in an environment dominated by firms and developers that to some degree specialise in creating software, and whose incentives for doing so are usually pecuniary in nature. Some of these incentives may be driven by the potential profits to be earned from developing a new innovation, but other goals may be realising lower production costs or increased efficiencies in production and delivery. Some profits are tied to sales of the product or hardware device containing embedded software, while other firms may derive profits from the sale of advertising, consulting, maintenance or other related services.

Non-monetary incentives may also be present, though usually these motivations yield some potential long-term economic advantage. For instance, an individual may gain experience and showcase his or her skills as a developer, particularly if operating in an open source environment. A firm might enhance its reputation by pioneering some type of discovery, even if it does not immediately yield economic returns. For example, a firm might develop a technique that makes it possible for disabled individuals to better utilise computer technologies, even if the commercial exploitation of this development comes years afterward. In such a case, the firm might lay claim to the intellectual property rights that underlie this discovery, even though the research and development does not yield an innovation, which is broadly defined as the successful commercial introduction of a new product or service.

Software innovation through different business models

Within the software sector, there are both proprietary and open source models. These models are not firm-specific, as one single company can employ various models depending on its needs. *Proprietary models*, in full or in part, seek to protect some aspects of the firm's source code by relying on strong patent and copyright protections. Doing so captures the monetary rewards that accompany successful innovation. Firms may chose to license some technology to other firms, or otherwise make technology available to outside developers in the interests of promoting the development of the ecosystem in a technology area, interoperability, or a common approach to a particular technical challenge, among other reasons. The open source model represents a combination of multi-party development and licenses that have varying levels of restrictions on commercialisation of the software. The theory behind this approach is that innovation is fostered when a variety of developers have access to the basic source code. Developers co-operate under conditions which take advantage of their peers' knowledge and skills, often drawing on a global base of developers. A firm benefits by letting users have access to its intellectual property, and gains developmental input that it would have had to generate on its own. The clear trend is for firms to adopt a hybrid approach that involves both proprietary and open source models, as they craft approaches to development and commercialisation that reflect their ever-evolving business models.

Collaborative activity is an increasingly important feature of business models and innovative strategies for software firms. Software development often builds on existing products and previous innovations. Due to the technologically heterogeneous and complex nature of software functionality, there is an increasing emphasis on collaborative approaches to developing improved and innovative functionalities that are interoperable. The diversity in software content and technologies means that individual firms or developers face challenges in delivering comprehensive solutions and generally must draw on resources beyond the firm in order to assemble the necessary elements for success. Some developers focus on their comparative advantage in delivering part of the process, such as product testing or marketing expertise. For software firms, collaboration draws upon external resources to deliver desired functionality and generate revenues.

Table 0.1. R&D investment by top companies operating in the fields of software, computer services and Internet, 2005

		R&D investment		Net sales		Employee		R&D intensity	
Company	Country	EUR (millions)	CAGR 3 years (%)	EUR (millions)	CAGR 3 years (%)	Number	CAGR 3 years (%)	R&D as % of sales	R&D per employee (EUR thousands)
Software									
1. Microsoft	USA	5 581.52	12.2	37 540	11.2	71 533	9.2	14.9	78.0
2. Oracle	USA	1 586.97	16.6	12 191	14.9	56 133	11.4	13.0	28.3
3. SAP	DEU	1 088.63	6.2	8 512	4.7	34 550	5.3	12.8	31.5
4. CA	USA	662.09	3.5	3 226	6.9	16 000	0.0	20.5	41.4
5. Symantec	USA	578.27	51.2	3 513	43.3	16 000	55.0	16.5	36.1
6. Cadence Design Systems	USA	358.88	3.0	1 127	0.9	5 000	-1.1	31.8	71.8
7. Adobe Systems	USA	309.70	14.1	1 667	19.1	5 734	19.7	18.6	54.0
8. Sega Sammy	JPN	298.73	n.a.	3 704	n.a.	5 407	n.a.	8.1	55.2
9. Intuit	USA	285.34	14.4	1 727	14.4	7 000	2.5	16.5	40.8
10. Synopsys	USA	271.27	12.1	841	3.0	4 756	3.8	32.3	57.0
Computer services									
1. IBM	USA	4 559.15	4.3	77 258	3.9	329 373	1.4	5.9	13.8
2. Unisys	USA	330.28	-1.9	4 882	0.9	36 100	-0.3	6.8	9.1
3. SunGard Data Systems	USA	207.53	12.0	3 318	15.7	15 000	19.5	6.3	13.8
4. DST Systems	USA	110.88	-3.7	2 132	1.8	10 500	-3.5	5.2	10.6
5. Indra Sistemas	ESP	85.90	6.0	1 202	11.2	7 584	7.6	7.1	11.3
Internet									
1. Google	USA	508.23	145.5	5 204	140.7	5 680	n.a.	9.8	89.5
2. Yahoo!	USA	498.07	56.0	4 457	76.7	9 800	39.6	11.2	50.8
3. Check Point Software Technologies	ISR	42.85	20.7	491	10.7	1 414	5.5	8.7	30.3
4. United Online	USA	33.92	17.3	445	46.3	900	28.9	7.6	37.7
5. F5 Networks	USA	26.58	20.3	239	37.5	792	n.a.	11.1	33.6

Notes: n.a. = not available; CAGR = compound annual growth rate. The table refers to the top companies in terms of expenditure on R&D.
Source: European Commission (EC, 2007), *The 2006 EU Industrial R&D Investment Scoreboard*.

The software sector innovation environment

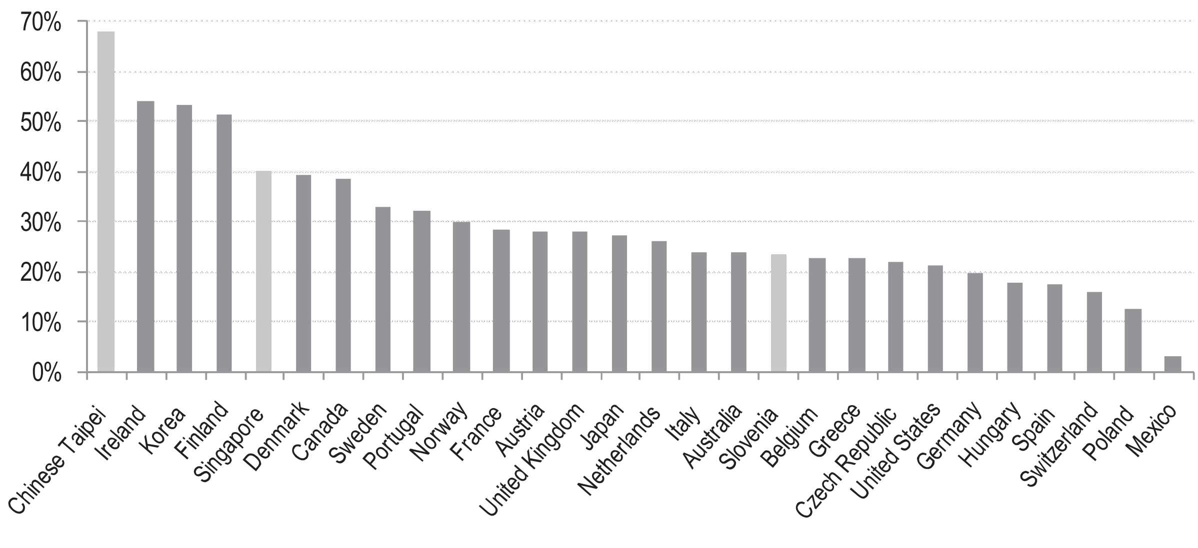
The intensity of software-related innovation activities is influenced by a variety of factors that determine its success and that these conditions vary from one place to another. The main factors, highlighted by recent OECD research, include the research and development intensity, the quality of the intellectual property rights regime and the availability of human capital.

R&D investments

A defining characteristic of the software sector is that the final product is highly dependent on research and development (R&D) that may result in, but does not always yield, an innovation, defined as a successful commercial launch of a new product or a significant improvement of an existing product. Yet once a software product is fully developed, it can be replicated and delivered at relatively low additional cost. In the software sector the main costs of developing products occur in the R&D phase.

As shown in Table 0.1, R&D expenditures by the top commercial firms producing software, computer services and Internet services are massive, and for some the investment growth has increased substantially, as gauged by a three-year compound average growth rate in R&D investment. This table shows the strong presence of US firms, but this data does not reflect the geographic location of the R&D activity, a point that will be amplified later. For now, it is important to note that the share of ICT-related employment grew substantially over the last decade. Because the data on software is poorly captured by existing economic statistics, taking these investment and employment numbers together gives some indication of the growing importance the software-driven ICT sector has in economic activity. Figure 0.2 shows the scale of human resources deployed in ICT across 25 developed countries, with the United States accounting for about 50% of the total.

Figure 0.2. Share of ICT R&D researchers in total R&D researchers, 2006 or latest available



Source: OECD (2008b), *OECD Science, Technology and Industry Outlook 2008*, OECD, Paris, *www.oecd.org/sti/outlook*.

Human resources supplies

The ICT sector is highly dependent on human resources. Rapid technical change and expanding market opportunities mean that ICT firms, including software firms, are constantly looking for new types of skills. The sector cuts deep and wide across the economy, which makes it highly dependent on the availability of talent that is both specialised and versatile. The supply of talent to the software sector has been and continues to be a major challenge to most OECD economies. The results from the OECD business questionnaire used in this study (see Box 1.1 in Chapter 1) indicate that firms tend to view availability of trained human capital as a crucial factor for software development. There is a strong need for governments to enhance formation of skilled human capital and to enable cross-border flow of high-skilled IT talent, especially software talent.

Human resources are behind much of ICT activity clustering. ICT firms have tended to cluster in concentrated geographic locations that aggregate this human capital. To some extent, this clustering has been influenced by government funding centres near particular research universities.

IPR regimes

Finally, software innovation processes are particularly sensitive to the quality of the surrounding IPR regime. Software-related innovations can be protected as intellectual property and, as is often the case with intellectual property in other fields of technology, remain vulnerable to imitation. To maintain incentives to innovate, governments have developed a variety of means to protect the rights of innovators in the software sector. International agreements, either bilateral or multi-lateral, provide a basic framework for the protection of IPRs in the software sector, including in some cases standards for minimum protection.

The specific mix of protections available varies among countries, but copyright protection is generally afforded to software innovators, as is some form of patent protection for software-implemented inventions. Other types of protection are also available to software innovators, such as protection for trade secrets or trademarks. A broad spectrum of software innovators has come to rely on IPR protection as an integral part of their business strategies; this includes most producers of both proprietary and open source software and firms or individual programmers operating under a wide range of business models.

Under current international arrangements, copyright protection for software innovation is quite broad and in theory available in countries representing the vast majority of the world economy. Members of the World Trade Organisation (WTO), with 151 member economies at the time of writing) are subject to the Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPS), which provides for computer programs to be treated as literary works, eligible for protection under the Berne Convention for the Protection of Literary and Artistic Works (administered by the World Intellectual Property Organisation, WIPO). Under these accords, copyrights automatically come into effect with the creation of the work and generally benefit from minimum standards of protection and national treatment in other signatory countries. Signatories commit to recognise and enforce this protection internationally.

Concerning patents, a world-wide comparison of patents issued for software-related inventions shows that to some degree firms will locate their R&D activities in nations with strong intellectual property rights and sound legal institutions. One important aspect to note from patent data is that the bulk of software-related patent applications are made by firms whose primary business activity is not software development. As software becomes more integral to a wider variety of products, often embedded in products, such as automobiles, which in the past did not use software, the range of software applications will continue to expand. The importance of effective intellectual property regimes is therefore of importance to a growing cross section of industry.

Some observers consider patents and copyrights as complementary tools for the protection of intellectual property with respect to software (Einhorn, 1990): copyrights protect original computer programs against unauthorised copying, whereas patents can be used to protect inventions (especially the underlying technical ideas and principles). Key differences arise in the characteristics of copyrights and patents: copyrights protect the work internationally without formalities, whereas patent protection is only granted in countries where the rights holder has applied for a patent (in some economies patents cannot be applied to software in certain forms; *e.g.* the European Patent Convention excludes patents on programs for computers as such). In practice, the duration of protection for a copyright is longer (generally amounting to 70 years or author's life plus 70 years), whereas a patent usually expires and gives the owner protection over 20 years from the date of application.

Emerging trends, changing environment and growing challenges

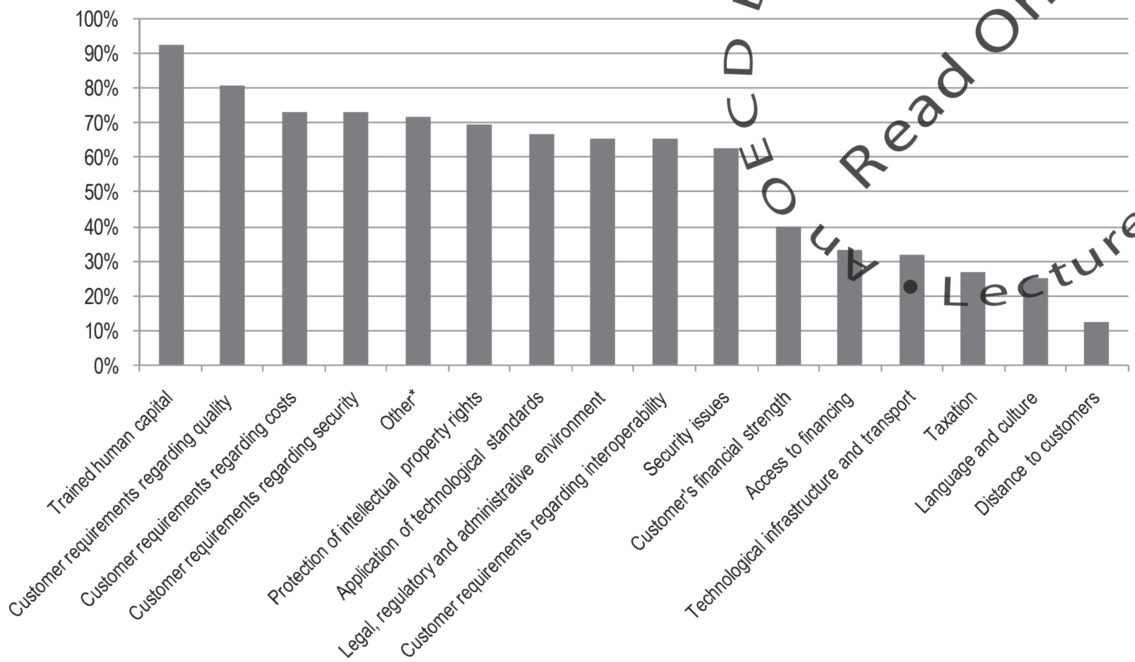
Globalisation realigns the market environment for software innovation

The previous section provided strong evidence that the market environment for R&D in the software sector is characterised by distinct geographic advantages tied to human capital, public investment in R&D, and strong legal protection of intellectual property rights. Yet "the death of distance" presaged by the growth of broadband technologies is realigning the nature of activities in the software sector along more global lines. Multinational enterprises are diffusing their R&D to take advantage of global technology sourcing. Emerging markets are playing an increasingly important role in this development, and while early evidence showed that the only lower-end services were being offshored, there is increasing evidence that higher-end activities are also increasingly being located in some emerging countries. Wage differentials may drive the locational decisions for some activities, but factors such as better human capital, intellectual property protection and the local business environment will be the predominant determinants for higher-order activities.

Figure 0.3 is drawn from the result of an OECD questionnaire, which reviewed the importance of various factors for software development activities of companies. The results stress the importance of some ICT infrastructure, human capital, legal institutions, and overall R&D environment.

Figure 0.3. Factors of high importance for innovation, as assessed by software developers

Percentage of replies; respondents could indicate multiple factors



* Other factors identified by developers.

Notes: N = 26. The figure presents the share of firms identifying each factor as being of high importance to the firm's software development operations.

Source: OECD (2008a), business questionnaire and related responses.

Evolving end user requirements for software functionalities

As society in general comes to depend more on ICT and other products which are driven by software to conduct civic, economic, and social activities, user-centred functionality requirements will increasingly guide software and other ICT innovations. And users themselves – individuals, firms, and governments – will play a growing participatory role in driving this innovation. Software's intrinsic nature as a digital, intangible product enables the real-time global transmission of content and the incorporation of user inputs into existing products. For instance, it is common for software developers to circulate versions of products undergoing development or refinement to solicit feedback in the final stages of development before the software is widely released. This means that users are contributing to software development in collaborative ways that do not take place in other industrial sectors. The software sector is a leader in engaging users to help develop and innovate software functionalities. Five of the most important functional considerations are reviewed here: *i*) mobility, *ii*) interoperability, *iii*) accessibility, *iv*) security and privacy, and *v*) reliability. Yet it is important to note that, in many cases, the specific functionalities are not an end but rather a means to obtain a desired aspect of performance from a software product.

It is also essential to recognise that market demand for software functionality plays an important role in propelling technological innovation by providing signals and incentives for innovators to act. At the same time, the nature of software as a digital, non-rival product means that there can be large returns to scale for innovators that are able to respond to this demand. Software innovation is substantial within firms and across firm boundaries (see below). There is also considerable innovation in software distribution models and business models. Software can thus act as both a driver as well as an enabler of innovation.

Moreover, due to the technologically heterogeneous and complex nature of software functionality, there is an increasing emphasis on collaborative and, in some cases, open innovation[1] approaches to development of improved functionality. This is because the diversity in the content and in the technologies means that individual firms or developers face challenges in delivering comprehensive solutions and generally must draw on resources beyond the walls of the firm in order to assemble the necessary elements for success. The contribution of user knowledge and experience to the development of software functionality has emerged as an important source of input for innovation processes; the software sector is a leading sector in the engagement of users, a factor that is contributing to the dynamism of the innovation activity.

Growing demand for functionalities

As noted before, the range of functionalities sought by users is broadening and the exigencies for performance with respect to the various functionalities are increasing. An illustrative set of software functionalities is considered to address the user perspective. This set highlights examples that have become priorities for many users as well as the manner in which the software sector has innovated to deliver them. These include security and privacy, mobility, interoperability, accessibility and reliability.

Mobility

In terms of software-driven products, *mobility* refers to a computing structure that is available at any time and at any location. Enhanced mobility has become a key functionality sought in the market.

Mobility is fuelled by technological progress (*e.g.* broadband infrastructure and technological convergence in devices) that in turn fuels demand for further innovation (*e.g.* increased communication speeds). Improved broadband wireless networks enable delivery of digital content, such as music and television programmes, over a variety of different platforms and devices. Traditional stand-alone technologies are increasingly integrated – voice telecommunication is now delivered over computers, while traditional broadcast content is digitally delivered to a computer or a mobile telephone. Platforms thus are more mobile regarding points of access to the fixed infrastructure. In terms of software, mobility encompasses the physical mobility and connectivity of different devices, and non-physical considerations such as the product's ability to function in different environments.

Advances in platform software, middleware, embedded technologies and telecommunications infrastructure are enabling a wide range of vehicles for delivery of computing technologies to users. Smartphones offering voice telephony also have processing capabilities equivalent to that of personal computers in the 1990s. Mobile devices, such as cell phones, PDAs, and handheld electronic games, have small screens. Firms are working

1. The term "open" does not necessarily imply that the innovation is handed out for free or without an expected economic return.

to develop software that allows a fuller range of Internet browsing on these small-screen devices. The growing demand for mobile content provides a powerful incentive for software developers to innovate products and services that meet this demand.

This rapid growth in mobile content and communications has significant implications for firm-level labour productivity. Many businesses throughout the economy are working on mobilising their workforce, and this fosters trends that enable employees to work off-site or to telecommute, sometimes from very great distances. Mobile enterprise applications now go beyond e-mail and work on delivering remote access to company documents, collaboration, video conferencing, and other communications that require secure networks. In such an environment software must be flexible, and not designed for a specific set of fixed hardware devices. Since software architecture needs to be increasingly adaptable to a variety of devices, the ability of cloud computing and SaaS to offer platform-independent mobile access means that these technologies will likely become more prevalent in the coming years.

Interoperability

According to the International Organization on Standardization, interoperability is "…the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the users to have little or no knowledge of the unique characteristics of those units." It means that interoperability aims to achieve the harmonious working of heterogeneous software products and services that make up the ICT infrastructure, but the needs for interoperability extend beyond this sector. The software industry is witnessing an evolution where systems are built in a mixed environment using propriety and off-the-shelf software that is often integrated with legacy systems, custom software, embedded software and online services. As the reach of particular software products extends across various industries and applications, inter-operability has implications for how the larger economy functions. In order to grapple with this increased technical heterogeneity, users and developers need to better understand the role of interoperability, and to collaborate on realising the benefits interoperability can offer all stakeholders.

Interoperability is demanded by users especially in ICT-intensive sectors (*e.g.* aero-space, mobile telephony, petroleum, pharmaceutical and automotive) At the level of software developers, not every component is designed by a single individual or firm. If the products being developed can interoperate with other products and services, this can increase market demand for a firm's product. Cost-effective efficiencies can be realised that benefit consumers by lowering prices and decreasing the need to invest in new equipment or commit to a specific technological product or service. Business and government activities rely heavily on information technologies, and their investment decisions may be influenced by the degree of interoperability offered.

Despite the greater complexity and variety of software systems and new technologies over the last decade, software interoperability has improved. Many leading software firms are co-operating to align their technologies to achieve interoperability between their products. Individual firms as well as industry associations and international consortiums have launched initiatives to address interoperability issues.

Interoperability can be achieved in various complementary ways, including *i*) industry-community partnership and collaboration, *ii*) product design and testing, *iii*) sharing of technology and access to IP, and *iv*) implementation of technology standards. The efforts advancing are at collective level (*e.g.* sector), firm level, and government level (*e.g.* guidelines).

Yet while concerns about interoperability strive to harmonise various technologies, there are different views regarding how to achieve this goal. While some technological standards must be shared, there is concern that potentially innovative technologies could be discouraged. Thus there is a need to maintain technological neutrality and choice, particularly with respect to government-issued guidelines concerning interoperability.

Accessibility

Accessibility is an important area from a social perspective: developments such as voice recognition software or keyboard enhancements can help promote social inclusion. Software-related technologies play a large and ever-expanding role in the modern civic and economic life present in OECD member nations. In recognition of this, accessibility is concerned with ensuring that as many members of society as possible can benefit from using computers. Software interfaces can be designed to overcome barriers faced by those with disabilities or age-related impairments. For those with visual limitations, screen colours and font sizes can be augmented to help those with poor eyesight, including blindness, and those who lack the ability to perceive certain colours. Keyboard devices help improve navigation functions, while comprehension software may help those with dyslexia or other cognitive limitations see and hear the text as it is manipulated on the computer screen. Voice recognition software allows a person to simulate keyboarding inputs or using a mouse by speaking into a device that directs the computer to perform these functions.

Signals to innovate in the area of accessibility come from a variety of angles including, among others, consumer demand. The software innovation in this area has also been encouraged by government mandates and procurement, but ongoing technological changes mean new opportunities to enhance operability will continue to arise.

Security and privacy

Security and privacy are distinct but related issues in the area of information technology. A paramount consideration for all stakeholders is ensuring the security of information technology systems and respect for the privacy of individuals who use such systems. Software innovation that improves *security and privacy* benefits both individual consumers and the larger economy, where commercial transactions are increasingly conducted on the Internet. Online personal and financial information can be stolen, while malicious software viruses and other programs can compromise or cripple computer systems.

Software determines what a system does with data, including how it is collected, stored, processed, linked with other data, shared with others, and how long it will remain in the system. From a developer's perspective, innovations in software security are about identifying, implementing, and operating the technical measures that will control and limit the security of an information technology system to an acceptable level. Innovations in the area of privacy include tools that maintain the integrity and dissemination of user information and that allow users to manage what is done with that information. To reduce risk, software programs include security and privacy features, such as password protection or secure coding. Yet software security controls need to operate in concert with other security management strategies, such as administrative controls that manage how the system operates, and physical controls involving alternate power sources or data backup devices. Users will balance these security and privacy requirements along with other desired functionalities, such as mobility and interoperability. As software shifts from a product to a service in some areas, as Internet connectivity and mobility are increasingly global, and as online crime is a new reality, users are increasingly risk averse when it come to software security issues. Innovation regarding information security and privacy is particularly important for its ability to safeguard confidence in the Internet-based economy.

In sum, security and privacy can be complex functionalities to implement; they have technical, administrative and physical dimensions, but also relate to user education and behaviour.

Reliability

Software reliability is part of a cluster of inter-related concepts linked to the ability of users to have confidence that the software products they employ will work consistently and not in an unpredictable manner. With the world becoming more and more dependent on software, software failures can cause more than mere inconvenience. Today, software errors or failures can expose society to the risks of critical infrastructure failures, severe economic loss or even human fatalities. Where they occur, reliability problems can originate from a variety of sources ranging from poorly designed user interfaces to direct programming errors or improper implementation. In this respect, software programs need to provide a predictable level of reliability and dependability. Stakeholders will establish different measures of reliability, but for some settings, such as air traffic control, automotive safety, or nuclear energy, no failures are tolerated. The complexity of the programs and the environment ensures that achievement of reliability is a clear challenge and efforts to address reliability need now engage the full range of stakeholders including developers, vendors, users and others.

As software's importance in the economy expands, so has demand for reliability. The market is responding to this demand. The demand for reliability is fulfilled in various ways, including automated methods and human analysts. For example, black box testing implements the software without requiring that users understand its internal structure, while white box testing lets the examiner take the internal data structures, codes and algorithms into account when evaluating the software. Alpha and beta testing are also used to test software reliability. Part of the challenge is evaluating software programs in various domains, and such testing procedures can be extremely labour-intensive. Many developers invest enormous amounts in efforts to ensure an appropriate degree of reliability. The cost of ensuring reliability is often directly related to the cost of testing, which can be very expensive. Yet failing to adequately address reliability issues can result in product recalls and costly interventions to deal with shortcomings. Many software developers solicit user feedback even after software is released, and this information helps improve future versions of the product.

References

Ahmad, N. (2003), "Measuring Investments in Software", OECD Directorate for Science, Technology and Industry Working Paper 2003/6, OECD, Paris, available at *www.oecd.org/sti/working-papers*.

Chesbrough, H.W. (2003), *Open Innovation: The New Imperative for Creating and Profiting from Technology*, Harvard Business School Press, Boston.

Chesbrough, H.W., W. Vanhaverbeke and J. West (eds.) (2006), *Open Innovation: Researching a New Paradigm*, Oxford University Press, Oxford.

EC (2007), The 2006 EU Industrial R&D Investment Scoreboard, European Commission, Joint Research Centre, Directorate General Research, Technical Report EUR 22348 EN, *http://iri.jrc.ec.europa.eu/research/scoreboard_2006.htm*.

Einhorn, D.A. (1990), "Copyright and Patent Protection for Computer Software: Are They Mutually Exclusive?", *The Journal of Law and Technology*.

Grimm, B., B. R. Moulton, and D. B. Wasshausen (2002), "Information Processing Equipment and Software in the National Accounts", NBER Working Paper (prepared for conference on Measuring Capital in the New Economy), available at *www.nber.org*.

OECD (2005), O*slo Manual: Proposed Guidelines for Collecting and Interpreting Technological Innovation Data, 2nd edition*, OECD, Paris.

OECD (2006a), *OECD Information Technology Outlook 2006*, OECD, Paris, *www.oecd.org/sti/ito*.

OECD (2006b), *Innovation and Knowledge-Intensive Service Activities*, OECD, Paris.

OECD (2008a), "Innovation in the Software Sector: Business Questionnaire" and associated responses.

OECD (2008b), *OECD Science, Technology and Industry Outlook 2008*, OECD, Paris, *www.oecd.org/sti/outlook*.

Parker, R. and B. Grimm (2000), "Recognition of Business and Government Expenditures for Software as Investment: Methodology and Quantitative Impacts, 1959-98", US Bureau of Economic Analysis Working Paper, available at *www.bea.gov*.

Von Hippel, E. (1988), *The Sources of Innovation*, Oxford University Press.

Von Hippel, E. (2005), *Democratizing Innovation*, The MIT Press, Cambridge, Mass.

Issues for Policy Makers and Industry to Consider

The bulk of this study was completed before the current global economic crisis struck and hence the priority level of the policy issues addressed below might have been altered by the situation. Nevertheless, the policy issues raised in this study refer to long-term challenges that face the software sector in general and across countries. The report's findings confirm the twofold importance of innovation in the software sector – as a highly innovative sector in its own right and as a crucial component of innovation processes in other industries. Moreover, the software innovation processes occur along a large variety of business models employed by software developers from various sectors across the world. Software's increasing importance to economies leads to a question about its potential for public policy. The unique characteristics of software also call for special attention from policy makers.

The findings presented in this study and summarised in the foregoing led to the formulation of a series of issues for policy makers and industry to consider. These issues indicate the main barriers to the software innovation process and identify the main areas for policy intervention to foster innovation in this sector. These recommendations draw on current literature, OECD and other data, input from an advisory expert group, the results of an OECD business questionnaire (see Box 1.1 in Chapter 1) and other resources.

Skilled human capital remains the most crucial factor for software innovation

The software sector lies at the heart of the modern economy, serving as a driver, an enabler, and a diffuser of innovation across all sectors and industries. The sector cuts deep and wide across the economy, which makes it highly dependent on the availability of talent that is both specialised and versatile. The scarcity of skilled human capital is seen as a key challenge for the sector. This is often the result of either the lack of sufficient supplies (*i.e.* quantity), or the lack of adequate skills (*i.e.* quality), and sometimes both.

Governments have often responded to this human capital challenge by increasing and expanding ICT training opportunities as well as easing immigration legislation to facilitate access to an international pool of software and ICT trained workers. In parallel considerable outsourcing has taken place, both nationally and internationally. The impact of such public interventions has been mixed whereas the quantity of ICT trained workers in the OECD area has indeed increased; the dynamic nature of the demand has meant that there were always specific types of skills in short-supply.

But there are a number of things that industry and policy makers may consider to do in order to address the human capital challenge more adequately. These are outlined below:

- *Improve the versatility of the skill base of software developers.* The growing demand for software solutions across sectors and industries poses two particular challenges to software developers. Firstly, it requires software developers to respond to a continuous flux of technical challenges, and secondly, it forces them to adopt different, flexible human resource sourcing models for different market segments. In other words, there is always the twin demand for new technical skills and new solution delivery models. It is therefore critical that there are enough graduates to satisfy a constantly changing demand for different types of skills. This is something that can only be achieved by improving the versatility of software graduates, and the flexibility of the software industry workforce.

- *Increase the supply of graduates with software development skills by embedding 'software development' training across various curricula.* But the acquisition of software development skills need not be confined to graduates of ICT studies only. From business and finance to manufacturing and arts, software plays an increasingly critical role in achieving business objectives. While many higher education institutions (HEI) in OECD member countries have integrated various aspects of software development training in non-ICT programmes (*e.g.* computer programming for humanists), there remains a lack of sufficient recognition of the importance of software training outside ICT industries and its role in enabling innovation within organisations and across the economy as a whole. The increased role of users in developing and stimulating software solutions (innovation) makes it imperative that all graduates have some understanding of software issues. In principle, every HEI should consider establishing a software development competence centre focused on software issues. The embedding of software development skills across a broad range of specialisations will help ensure that software development skills are more widely spread in the workforce and are not confined to actual software workers per se (*i.e.* those whose work revolves primarily around the development of new software).

- *Ensure the "training of the trainers".* The high competition for IT workers, including software developers, over the last decade or so has meant that universities and colleges have had to compete fiercely to attract and retain highly skilled IT workers. The disparity in pay and work conditions between public research centres and higher education institutions on the one hand and the business sector on the other has meant that the former were less able to compete for talent with the latter. In the long run, both sectors will lose as public research and academic institutions will suffer from a shortage of 'IT scholars' and firms will suffer from the shortage of 'IT workers'. This calls for the particular attention of policy makers to increase funding opportunities for software research at public research organisations and higher education institutions in order to create more and better career opportunities for software researchers in the not-for-profit sector. This is important to prevent a potential gap in these areas which could eventually result in the lack of advanced highly skilled trainers, which in turn could spill-over to other parts of the human capital formation system.

- *Broaden the training base of software developers beyond technical skills.* The complex and ubiquitous nature of software implies that software workers should be able to operate in more than one market segment and should be able to quickly adopt new sets of technical skills as they become in demand. Consequently the training process of software workers should rely on human capital with general and broad background, capable of understanding the fundamental issues of software processes. The versatility and flexibility of software developers should go beyond the sets of technical skills they may possess. Software developers add value to the economy by creating solutions to problems, and responding to needs, emerging from a variety of industries and sectors. It is important therefore to incorporate interdisciplinary training into software development programmes, especially in business, operations, and communication skills, but also in social sciences, humanities and the arts. This will help produce software talent capable of understanding the business needs of different industries and capable of communicating with users from a broad range of sectors. More important, this will help diffuse software workers across the economy and help create further opportunities for (interdisciplinary) innovation to emerge on the basis of software skills and platforms. In this respect, the diffusion of software skills can be a major enabler of innovation across the economy.

- *Build and strengthen management competences at software firms, especially at SMEs.* Software firms are constrained by management capabilities, which require strong science and technology knowledge as well as entrepreneurial skills. In this context, a key factor to foster software innovation is the leaders of software developing firms, skilled in management and able to pursue new business solutions. Thus, a well-performing and broadly accessible education and training system for managers will facilitate the adoption and diffusion of innovation. Balanced and collaborative programmes aimed at addressing shortages in management skills, technology and human resource development can make an important contribution in upgrading SME skills.

- *Improve and widen channels of collaboration and communication around human resources issues between academia and industry, especially with SMEs.* The creation of a flexible workforce will require greater co-operation between industry and academia, particularly in the development of curricula but also in the provision of practical training. Linkages and collaboration between the academia and industry need to be further developed, with the aim of improving knowledge diffusion and prepare students for entering on the job market. These linkages could range from formal business-university collaboration to informal interactions and partnerships (*e.g.* through trainee programmes). The informal interactions, personal contacts and networks between academia and other organisations are critical, but tend to be mostly outside the policy scope. Nevertheless, the tertiary education sector should be flexible and responsive to software industry needs in terms of co-operative projects.

- *Policy needs to ensure that small and medium-sized enterprises (SMEs) from all software-related technological sectors are considered when programmes or co-operation with academia are designed.* This is particularly important given that SMEs developing software reported considerably less co-operation with academia. Moreover, some existing linkage programmes are largely suited to longer-term arrangements, and this may hinder participation by small firms.

- *Remove barriers to effective mobility of human resources.* While the demand for new graduates in software-related fields is high, local supply may be not sufficient to deal with shortages, and mobile (foreign) human capital needs to be accepted as a core part of the software industry and its innovation processes. In this context, many OECD governments have designed skilled migration programmes to improve the inflows of foreign talent. However, the cost of labour relocation and mobility can be very high and new forms of access to international talent have emerged as a result, including outsourcing and offshoring of jobs. OECD governments should contemplate new forms of short-term 'high-skilled worker visa' to improve the rapid and temporary flow of needed highly skilled workers between countries. It is important that short term talent flows are distinguished from long term talent migration and follow different and more rapid procedures.

Technical infrastructure and environment

Generally software development draws on the existence of hardware (*e.g.* a PC, or software embedded in a car, or in a TV set) and increasingly on the existence of a strong communications infrastructure, particularly broadband and wireless networks (*e.g.* for applications running on mobile telephones or personal digital assistants). Equally important is the existence of legal frameworks (such as technical standards and intellectual property rights) that are conducive to innovation. Government policies can impact significantly on software innovation through its role in the provision of both physical and legal infrastructures that combined have the potential of providing the environment needed for the software industry to flourish.

- *Further boost broadband capability and wireless networks.* The interplay of software, hardware and telecommunication technologies in advancing technologies of various software functionalities including mobility and accessibility, highlights the importance of network infrastructure. Continuous investment in broadband infrastructure, particularly wireless and fibre networks, is needed to meet growing user requirements and expectations. Facilitating the development of high-speed broadband networks will enhance upstream and downstream capabilities and open new venues for innovation to emerge. Governments have an important role to play in reducing barriers to market entry, facilitating the opening of wireless networks, and encouraging new network investment. Broadband networks, wireless or otherwise, are critical conduits of innovation in the software industry as they are both drivers and enablers of software innovation.

- *Recognise the potential benefits of technical standards in stimulating innovation.* The issue of technical standards is particularly relevant to innovation in the software sector. It is important that governments refrain from mandating particular technical standards based on whether the standard has been developed or adopted by an established standards setting body. Rather, governments should embrace a flexible and neutral approach with respect to competing technologies that allow for choice based on objective performance criteria that can drive the best possible technical solutions. Nonetheless, governments can facilitate the setting of technical standards when interoperability issues are at stake. Technical standards that facilitate interoperability and open networks help create a technical infrastructure that engenders a wider environment for potential software innovation incentivised by larger economies of scale.

- *Recognise multiple approaches towards the innovation process, particularly in the context of functionalities development.* As economic and social activities are deeply affected by software, issues relating to software functionality (such as mobility, security and privacy, reliability, interoperability, and accessibility) tend to differ across stakeholders. This calls on policy makers to include various stakeholders in the process of functionality-promoting policymaking. Moreover, user participation is needed as the contribution from user knowledge and experiences has emerged as a valuable source of input for open innovation process in the software sector.
- *Ensure that IPR regimes related to software innovation are efficient and balanced and enforce them properly.* Software-related innovations remain vulnerable to imitation. To maintain incentives to innovate in the software sector, governments have developed a variety of means to protect the rights of innovators. But for intellectual property rights to be effective they must be combined with effective enforcement, otherwise the practical value of laws and regulations is low. Increasingly, many aspects of software development are being shared through various platforms of collaboration and co-development. Policy makers together with main stakeholders (including academia, businesses, and users groups) need to work together to put in place IPR regimes that strike a balance between providing incentives for collaboration and rewards for innovation.

Recognising networking and spillovers

The software sector is evolving rapidly and its innovations are impacting on a broad range of market actors from various industries and across the world. This poses an additional challenge for policy makers who have to work with ever widening and changing contexts. Software innovations in one industry can have significant impact on many other industries and the geographic scope of such impact knows no boundaries.

Thus, the policy maker should not overlook the direct and indirect impact that software has on the productivity of the local economy and its competitiveness. For example, there can be more value created through services that rely on software than by the software directly. In other cases, the software is part of a product (*e.g.* embedded software) or provided as a service through Internet (SaaS). Already in 2003, in most OECD countries, ICT services (including a substantial share of software) were more than two thirds of the value added of the total ICT sector. The multiplying effect of software innovations on the economy are very large and go far beyond the economic value created directly by the industry *per se*.

- *Foster linkages and collaboration, including internationally.* Strong linkages and collaborations form part of the success in software innovation. As the software innovation process has become more global; ideas and knowledge for innovation are now drawn from many, often global sources, and linkages and co-operation are of growing importance for successful innovation. Internationalisation is exercised not only through foreign investment, multinational enterprises, and trade, but also increasingly through international communities of practice and international networks. This is true for both commercial and open-source software activities.

- *Promote networking to facilitate the formation of clusters.* As the flow of information and knowledge is essential in the software sector, clusters, science and technology parks, and development agencies can play an important role, especially in enhancing R&D capabilities of SMEs. Successful software firms tend to cluster to exploit high quality infrastructures and to tap into pools of talent attracted by clusters of firms, *e.g.* Silicon Valley in the United States or Bangalore in India. In networking various actors together, not only central government but also cities and regional governments should actively invest and involve in the process of clustering. Cities and regional governments can underpin and strengthen this function through partnership with SMEs, public institutions and universities. Technology clusters come in different shapes and forms. They can emerge around centres of research and learning, around large firms, in adjacency to industrial complexes, or through the spontaneous co-location of small firms attracted by certain incentives. The diversity of the software industry means that different firms will benefit from different types of clusters. For example, new start-ups will benefit from clustering around centres of research and learning due to what they can offer them in terms of access to wider set of resources, while more mature firms will benefit from clustering nearer to their main clients which will allow them to develop their products and services in close contact with the client. Governments need to recognise the differences between various types of clusters and types of firms along the value chain.

Policy coherence

The pervasiveness of software in various parts of the economy and in daily life means that the success of public policy in this arena requires a strong co-ordination among diverse stakeholders.

- *Encourage strong policy co-ordination.* Although much of software innovation activity is taking place under auspices of the software sector, it is important to recognise significant contributions from other sectors. The public sector can influence the environment for software innovation in both indirect and direct ways, such as through promotion of human capital development, fostering the development of Internet infrastructure or the promotion of e-accessibility and security. A strong contribution also comes from industries beyond the traditional software sector as many of these are heavily engaged in developing embedded software. Much can be achieved through strong co-ordination between various agencies, international standards bodies, ministries and private companies across different sectors.

- *Align policies that impact on software functionalities.* The increasing computing capabilities together with the presence of multiple communication technologies are building a new infrastructure in which software functionalities (such as mobility, security and privacy, reliability, interoperability, and accessibility) are becoming more important every day. Different policy agendas for functionalities can sometimes lead to bottlenecks, trade-offs and eventually dysfunctionalities, especially from a user perspective.

- *Establish effective and timely evaluation of polices.* For public policy to be effective it has to be relevant and has it desired impact. It is therefore important that regular evaluation of policies is undertaken with the engagement of various stakeholders, and through wide public discussions. Policy makers should then respond to evaluation results and reassess their policies accordingly. Furthermore, commitment to evaluation at the outset could raise the level of expectation that evaluation will be fed back into policy development and will influence subsequent policy that reflects recent market and technical developments.

Framework conditions for SMEs

For most SMEs access to new knowledge and R&D activities is expensive and lies often beyond their means. SMEs also face many challenges in entering markets and accessing talent and capital. Generally they must draw on resources beyond the firm to assemble the necessary elements for success. Many small software firms reported encountering difficulties in accessing finance, building knowledge networks and recruiting specialist staff. Governments need to address those difficulties and provide efficient mechanisms to boost the innovation capabilities of SMEs.

- *Provide comprehensive market-based financial support mechanisms that focus on early stage SMEs.* A lack of appropriate financing represents a hindrance to the creation and expansion of innovative SMEs in the field of software. Comprehensive market-based approaches are needed to bolster the early stages of SMEs which are usually marked by negative cash flow due to high level of start-up activities that do not yield immediate returns. These approaches include better access to venture capital, guarantee fund, institutional capital and better use of related public funds, which can be achieved through application of various solutions, including development of sound venture capital market. For example, venture capital in Israel has played a pivotal role in helping start-ups bring innovative software products to the market place. Successful approaches to developing early stage venture markets include tax-based programmes and programmes that use government's ability to leverage private risk capital.

- *Support proactive management and use of firms' intellectual assets.* Software firms depend heavily on investment in intangibles such as R&D, customer relationship management and training. Often, these intellectual assets-intensive companies feature specific operational and business risks and this can have a major impact on the value of the companies concerned. It is important to be able to transmit intangible assets to others at crucial time of the business cycle. Therefore, reliable information about the intangible assets of companies is needed and the information needs to be disclosed to avoid inefficient resource allocation and facilitate innovation. Methods of measuring and disclosing intellectual capital should be developed, diffused and linked to the upgrading of financial services. Evaluation and developing comparable reporting of intangibles could make a valuable contribution in furthering innovation in the software sector.

Improving measurement and data collection

Software is often poorly captured in economic statistics because of at least three main reasons. First software is evolving rapidly to take on new forms (*e.g.* software as a service). Second, it is a non-physical good and a part of it is created and exchanged outside the monetary environment. Last, there are lots of cross-border innovation activities and exchanges with software products. In order to develop policies that support software innovation appropriately, it is necessary to better understand several critical aspects of the innovation process. More emphasis on improving of measurement and data collection can help strengthen the understanding of policy development for software.

- *Promote international co-operation, which is essential to harmonising definitions and data collection methods.* It is not always clear what software sector is and where the dividing line is between various measures within a software sector (*e.g.* consumption and investment). The decision what is the software sector or what constitutes investment in software can often be based on relatively subjective factors. This can give rise to significant measurement differences across countries. This is true for all products and not just software; however, given its intangible nature and other special characteristics (*e.g.* ”ownership” rules), software measurement is more sensitive to ambiguities concerning definitions. To improve the comparability datasets across countries, terms that determine data collection (*e.g.* software sector, or software investments) should be unambiguously defined, and various statistical categories related to software should be harmonised.

- *Implement methodologies that capture the broader software sector.* Software innovation occurs not only in the classically defined software sector (although this definition can vary across countries) but also in other industries, such as automobiles or consumer goods. Existing methodologies focus mostly on the classically defined software industry, which could lead to significant underestimation of the importance of software. Thus an extension of traditional data collection techniques is required to capture software development and software across various industries.

Moreover, much of software innovation occurs outside the classical monetary market, which it is related to by the non-physical nature of software. These activities are not captured by most statistics, however. It is recommended to work on the development of methodologies that capture all types of software, including those that are beyond the monetary market.

Chapter 1

The Economic Processes of the Software Sector

This chapter analyses how software (applications, software development tools, and systems infrastructure software, including embedded systems) differs from other technical fields in terms of innovation. It also explores the various economic processes – invention, production, distribution, use – for which software could turn out to be "different" – or, conversely – similar to other technical fields. In addition, this chapter discusses the ways software is measured by available statistics.

Introduction and main findings

Software has become a virtually ubiquitous phenomenon in modern economies, playing a vital role in a growing number of fields with respect to products and processes. There is a huge variety of possible functions that software can offer – from core codes and operating systems to applications that provide various functionalities to end users. Embedded software has become a key component of various products and systems. Business applications are important inputs in a growing number of industries.

Innovation in the software sector has twofold importance for OECD economies: software is not only a highly innovative and economically important sector in its own right, but it is often also an important element of innovation in other sectors. Many process and structural innovations depend heavily on organisational changes that are facilitated by software innovations. Such changes can be a major driver of productivity improvements. Thus, the expected economic impact of software innovation is likely to be much greater than what is observed by solely examining capital investment in the sector.

Moreover, software innovation has gone global. Technological developments, in particular in information and communication technologies (ICT), and the roll-out of high-speed broadband communications networks, have hugely increased the scope for globalisation of software activities. As a result, production and consumption of software can take place in different locations, internationalising software R&D activities and creating new market segments such as "software as a service" (SaaS).

Drawing on current literature, OECD and other data, input from an advisory expert group, the results of an OECD business questionnaire (see Box 1.1) and other resources, the Secretariat conducted this analysis over a number of months. The following assessment provides some important policy-relevant insights into the nature of innovation in the software sector, pointing to the following main findings:

- *Software is poorly captured by existing statistics*. Since it is a non-physical good and is evolving rapidly to take on new forms (*e.g.* software as a service), it is often poorly captured in economic statistics.
- *Available evidence points to a high degree of dynamism in the sector*. The software sector is dynamic in the pace and extent of its growth and innovation. The dynamism is reflected in the development of traditional software firms and the emergence of many innovative start-ups. It is associated with the emergence of new approaches to software business and interaction with partners in other industries, among other factors.
- *Numerous modes of software innovation are employed, some with software-specific features*. Innovation in the sector proceeds in an incremental and cumulative fashion, capitalising on the characteristics of software as an intangible, digital product. In some cases, traditional linear modes of innovation are employed; in others, collaborative approaches are employed (some of which are software-specific).
- *Business models in the sector interact with the innovation processes*. Some models are built around the innovative process (*e.g.* open source approaches), while others have been made possible by recent innovation in information and communication technologies (*e.g.* cloud computing). In some cases, the vendor-customer relationship resembles a strategic partnership, whereby development across the stakeholders is mutually reinforcing, yielding further innovation. The range of business models changes in an ongoing fashion.
- *Software innovation is human capital-intense*. The availability of trained and creative workers is a key element in the software innovation process; human capital is a crucial input. Depending on the nature of the innovation, the physical capital requirements can be relatively modest.
- *Environmental factors are important to the software innovation process*. The nature of the intellectual property regime, technical standards, legal and regulatory requirements and other environmental factors influence the ability of software firms to optimise their processes in favour of innovation. (The OECD business questionnaire used in this study provides some insights in this regard (Box 1.1).

Box 1.1. OECD business questionnaire

The OECD project on innovation in the software sector was carried out in co-operation with an advisory expert group (AEG) comprised of representatives from business, academia and government, with particular expertise in a range of relevant areas. With the assistance of the AEG, during the second quarter of 2008, the Secretariat circulated a detailed (20-page) questionnaire to businesses engaged in software innovation. Responses were received from 27 firms located in Germany, Japan, Lebanon, Mexico, Poland, Spain, Switzerland, Turkey and the United States, ranging in size from USD 1 million in turnover to multiple billions and from a few dozen employees to tens of thousands. Of these firms, about three-quarters were independent enterprises, with more than two-thirds having subsidiaries. About two-thirds had developing and selling software as their main activity. The business questionnaire was not intended to be statistically representative, but rather to provide illustrative insights from a diverse group of enterprises. Results are discussed throughout this chapter.

Taken together, these findings provide some useful pointers for consideration in the development of subsequent policy frameworks. They do not constitute recommendations. They do, however, point to both the need for caution in government policy so as to avoid interfering with the fruitful innovation processes, while at the same time delivering appropriate policy support in key areas of public interest (*e.g.* human capital development).

The nature of software

What is software?

Generally, the term *software* is used to describe the digital instructions and operating information that are contained in *programs* serving to guide machines – especially computers – in implementing desired operations (*e.g.* processing data or interacting with peripherals).[1] Software is created from source code, which consists of sequences of statements and declarations written in special computer programming languages. The source code is usually held in one or more text files, a large set of which may be organised into a directory tree (also called a source tree). Source code is usually converted into a machine-executable digital format using special compiler programs.[2] End users generally get only the executable files; the source code does not necessarily need to be revealed to the user. Open source software (OSS) either includes or permits ready access to the human-readable source code in order to facilitate further modification or reuse, subject to certain conditions (*e.g.* licensing).[3]

Two key features of software are that it is disembodied and complementary to hardware:[4]

- *Disembodied character*. Software is a non-physical good. It is a digital sequence of commands stored on physical carriers, such as a hard drive, memory disk or compact disk. Software can be recorded and stored in either a temporary or permanent fashion. The main implication of the non-physical form of software is that once developed, a software product can be replicated with relatively little cost and effort. Although a software supplier may incur some costs in delivery of products to the market such as direct cost of delivery (*e.g.* packaging), customer support or provision of software updates, the marginal cost of production for a given version of a software product is generally modest and may be close to zero in some cases. The disembodied character of software also facilitates digital transmittal (*e.g.* via the Internet), which can facilitate and expedite the delivery from the producer to users.

- *Complementarity with hardware*. Hardware provides the physical support for software and without it software is merely a set of instructions that cannot be executed by a machine. Similarly, modern computer hardware requires software to function and is typically incapable of providing any advanced functionality in the absence of software applications. As a result, there are strong linkages and complementarities between hardware and software markets. A given software product may be hardware agnostic, able to run on various types of hardware. Some software products, on the other hand, are especially designed for specific hardware products. The rapid pace of technological change in hardware[5] can influence software development; for example, programmers may move quickly to exploit expanding capabilities in new generations of hardware.

Such characteristics of software influence the profile of innovation in the sector including, for example, certain aspects of its nature and pace:

- *Cumulative development.* Most software development and change-in-added-value occurs through a multi-stage process that builds on previous innovative steps (FTC, 2003). The accessible, disembodied nature of software facilitates transfer, reuse and modification (*e.g.* revision, addition, linkages) of code and is conducive to such a process. Thus, cumulative development plays a particularly notable role in the case of software products, even though it is not wholly unique to the sector.
- *Short life cycles, especially for specific product versions.* The disembodied character of software and reliance on cumulative, incremental development processes (Harter *et al.*, 2000) together contribute to the relatively-frequent release of new products and product enhancements. As FTC (2003) points out, most software products experience much shorter lifecycles than other traditionally manufactured goods.

Software structure and function

Software is developed by many different entities including firms that specialise primarily in software, firms outside the sector that produce software in addition to their main lines of business, public sector and academic institutions, non-governmental organisations and individual programmers. The diversity of the stakeholder perspectives and the expanding nature of software itself, with deployment across most facets of life, make it difficult to propose one, unified and unambiguous taxonomy to classify its key dimensions.[6]

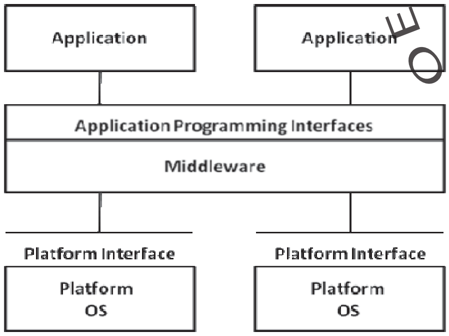
One approach is to sketch the various key types of software in terms of their function and relationship in the architecture and geography of systems. While there is bound to be some overlap in the various categories under such an approach (*e.g.* a particular function may be performed alternatively by more than one type of software, each in a variety of configurations), this mapping of software permits a general description and basic analysis.

Software architecture

Architectural approaches sometimes view software as a stack ranging from the machine interface to the user interface. While the architecture is changing and a multi-dimensional view is now required, software structure remains layered. One important dimension of change in recent decades can be found in the geography of software. While software still operates locally, it is increasingly structured to operate in a distributed or remote fashion.

A basic schema for software architecture considers three layers: applications, operating systems and middleware (Figure 1.1). Under this approach, software applications can be seen as programs that offer functionality directly sought by users, they may reside locally or remotely. Operating systems reside locally and manage hardware resources, serving as the software interface relating certain user requests to the hardware and providing the structure enabling other types of software to run. The third layer is middleware, which is employed in cases where processes run remotely. Middleware bridges among applications and operating systems by providing a standardised interface that can permit interoperation of applications on different platforms or written in different languages. Applications generally interact with other software via application program interfaces (APIs) that provide a means for them to communicate via specific language and message formats. These three elements of software architecture (applications, operating systems and middleware) are discussed in the following sub-sections.

Figure 1.1. One view of distributed software architecture



Source: Bernstein (1996), "Middleware: A Model for Distributed Services", Communications of the ACM.

Applications

Software applications are programs, usually designed to deliver sets of functionality to end-users and, sometimes, also to other applications. They are essentially located on top of the operating system and generally cannot function without the presence of an operating system to manage their interaction with the components of the hardware device where they reside. Common types of applications for household and office users include software for text processing, electronic spreadsheets, database management, instant messaging, email, web browsing and desktop publishing, among many others.[7] Specialised applications can also be specific to relatively narrow task sets (*e.g.* machine control in manufacturing) or to particular industry needs (*e.g.* film production software). Software applications are often bundled by developers in order to provide a consistent user experience across ranges of functionality; these are referred to as application suites. Within these suites, applications can interact with each other in order to facilitate data transfer and other user conveniences. Similarly, developers may use APIs to join simple applications together into composite applications or mash-ups that provide a greater range of functions in an integrated fashion. While applications may reside locally on the user's computer system, increasingly they are accessed remotely, including via the Internet; in some cases, they are offered as a service that is paid for based on usage or through a subscription fee.

Operating systems

In serving as the interface between users, other software and the hardware, operating systems perform a number of core functions (Silberschatz *et al.*, 2005). Operating systems:

- execute user programs,
- employ computer hardware in an efficient manner,
- bring convenience to user interaction with the computer.

They are the backbone of the processes that users demand, residing in many computing environments such as supercomputers, mainframes, servers, desktops, workstations, hand-held devices, real-time, and embedded systems. Examples from desktop computing may include such operating systems as Windows, OS X or Linux, among many others.

According to Silberschatz *et al.*, the specific functions of operating systems can be grouped into two large sets functions: *i*) those related to *process management* and *ii)* those related to *memory management*. Process functions involve actions such as process execution, management, supervision, interruption or destruction. Memory management involves supervision of hardware memory capacities. The latter function is important because programs function on a time-sharing basis and memory must be allocated in a manner such that programs do not unduly interfere with each other. Box 1.2 itemises the key activities of the operating system in more detail.

Box 1.2. Operating system activities

Management of the processor: allocation of the processor resource among the different programs using a scheduling algorithm, according to the desired objective.

Management of the random access memory: allocation of the memory space to each application and, where relevant, to each user. If there is insufficient physical memory (*e.g.* random access memory), the operating system can create a memory zone on the hard drive, known as "virtual memory". The virtual memory lets users run applications requiring more memory than is available on the system. There also can be cache memory where frequently executed instructions and data reside.

Management of input/output processes: integration and control of program access to material resources via drivers (also known as peripheral administrators or input/output administrators).

Management of execution of applications: allocation of resources required for applications to operate. An application that is not responding correctly can be "killed".

Management of authorisations: handling of security relating to execution of programmes by guaranteeing that the resources are used only by programmes and users with the relevant authorisations.

File management: reading and writing in the file system and the user and application file access authorisations.

Information management: provision of indicators that can be used to diagnose the operation of the computer.

Source: Kioskea, *http://en.kioskea.net/systemes/sysintro.php3*.

Middleware

Middleware, often referred to as business integration software, can be seen as a software intermediary between an operating system and applications which may reside on multiple machines. Middleware is used to facilitate application development by providing common programming abstractions, by masking the heterogeneity and the distribution of the underlying hardware and operating systems, and by hiding low-level programming details. Middleware development, production, distribution and innovation occur across a range of environments (*e.g.* with respect to the licensing approaches).[8] It permits applications to (Schreiber, 1995):

- locate transparently across the network, providing interaction with another application or service,
- be independent from network services,
- be reliable and available, and
- scale up in capacity without losing function.

Generally, the delivery of integrated services, which may be implemented across a diverse range of computing environments, is done in a fashion that is seamless in terms of end-user experience.

Concretely, middleware can enable systems to deliver services such as:

- distributed system services, including communications, program-to-program and data management services across a number of locations;
- application-enabling services, giving applications access to distributed services and the underlying network;
- management services, enabling applications and system functions to be continuously monitored to ensure optimum performance of the distributed environment.

Across the various systems and functions, middleware must operate securely, ensuring services such as communication encryption, user authentication and access control, as necessary.[9]

Beyond the basics

The basic categories listed above do not provide a comprehensive picture of the full scope of software and also mask many sub-categories of software within each of the headings. Beyond these basic categories, a variety of additional ones may be specified from a variety of perspectives. The following section describes several categories that are of particular interest from the perspective of software innovation within the scope of the OECD project.

Cloud computing

Cloud computing is a nebulous, software-based concept that goes beyond consideration of the functionality of the programs to include the manner of delivery, location and user experience, among other dimensions. Generally, the concept refers to application, platform or utility services accessed by users over the Internet in real time from the global ecosystem of service providers.[10] Cloud computing can be massively scalable, readily accessible, highly reliable, and cost effective, with low barriers to entry to new providers of services (Johnston, 2008). Often, the cost of using cloud-based services is comparatively low because cloud computing enables the providers to respond flexibly to market demand from a global pool of users for a particular software service, thereby exploiting economies of scale. Indeed, some advocates foresee availability of cloud computing power at utility prices, similar to electricity or water.

Although it is in an early phase of development, cloud computing is already permitting users to share resources in a generally fast and efficient way, accessing the extraordinary computing power available from large providers (Carey, 2008). It is a way to increase capacity or add capabilities without investing in substantial new internal computing infrastructure, software licensing and training. It can help firms reduce the need to acquire, configure and administer hardware and software, which can often be the source of project delays or failures. Generally, it can enable users to avoid the cost of down time or underutilisation that may plague some in-house systems. Often, the services are paid on the basis of usage (*e.g.* pay-per-use) or subscription, though some may be indirectly-paid via advertising or sales of third-party products.

For some software innovators, references to the cloud go beyond the delivery of software services to the market. The cloud is also a resource, whereby innovators can pose difficult questions to a large resource base (*e.g.* via blog) and draw on the collective wisdom among experts in particular internet communities to resolve them (OECD, 2008c).

Cloud computing is just now taking off for consumers, some of whom, for example, are turning to such sites as Google Apps (office suite) or Apple's MobileMe (storage and synchronisation services) for access to specific computing services. However, some enterprises are more advanced in the process of virtualisation as they strive to develop agile, scalable infrastructure. Others are turning to on-line platforms as a vehicle to develop and market applications (*e.g.* via Salesforce.com), which may enable them to meet their own computing needs while also potentially monetising applications they develop themselves through sales to other enterprises. Major players in cloud computing for enterprises include Amazon Web Services, IBM and Google, among others.

Though the concept of cloud computing is subject to further development, it generally encompasses such services as (Knorr and Gruman, 2008):

- Software-as-a-service (SaaS) providers deliver access to applications and other services via the Internet browser (discussed in more detail below).
- Platform as a service providers offer software development environments, as well as marketing and delivery channels.
- Web services in the cloud deliver application program interfaces (APIs, discussed below) that enable developers to focus on building the software functionalities they wish to deliver, while capitalising on off-the-shelf functionalities available from the service provider (*e.g.* such as transaction management).
- Utility computing constitutes a virtual resource pool available over the web for supplemental data storage, computing capacity, input and output management, among other functions.
- Managed service provision (MSP) refers to an application operating via the Internet and interacting with the enterprise user's locally functioning information technology processes rather than offering functionality to consumers. Examples of MSP include virus scanning service for e-mail or an application monitoring service.
- Service commerce platforms combine SaaS and MSP to deliver a service hub giving specific users access to controlled environments (*e.g.* with expense management in mind), that enable them to then purchase services such as travel or secretarial services.
- Internet integration refers to integration of cloud-based services such as might be offered to SaaS providers on a business-to-business basis or integrated solutions to consumers drawing on various SaaS providers.

Software-as-a-service

Under the concept of SaaS, the focus shifts from on-premises delivery to the customer of a software package towards delivery to the customer of a software service. As with the broader category of cloud computing, there is no standard definition for the concept of SaaS. Gartner (2007) describes SaaS succinctly as "software that is owned, delivered and managed remotely by one or more providers". A more expansive definition states that "Software as a Service is time and location independent online access to a remotely managed server application, that permits concurrent utilisation of the same application installation by a large number of independent users (customers), offers an attractive payment logic compared to the customer value received, and makes a continuous flow of new and innovative software possible" (Lassila, 2007).[11, 12] Although SaaS is still in a relatively early stage of adoption in the market, the influx of investment in infrastructure and entries into the market provide an indication that the SaaS model is gaining traction.[13] Table 1.1 presents an illustrative list of key differences between SaaS and more traditional software-as-an-application approaches.

Table 1.1. Traditional software-as-an-application vs. software-as-a-service

Software-as-an-application	Software-as-a-service
Users pay upfront for the license	"Pay as you go"
Dedicated instance of software is installed on user's hardware	Software is managed and maintained by SaaS provider
Users are responsible for deployment, operation and maintenance of the IT infrastructure required for the application	The SaaS provider is responsible for the infrastructure
Users are responsible for upgrading software	The SaaS providers upgrade software automatically

Source: Workday (2008), "From Applications to Services, the Shift to On-demand", White Paper, available at *www.cfo.com/whitepapers/index.cfm/displaywhitepaper/11778019*.

SaaS providers tend to have the following characteristics (IDC as cited in SIIA, 2001):

- *Application-centric:* SaaS providers generally offer access to, and management of, an application that is commercially available.
- *Sale of access to applications:* customers gain access to a new application environment without making up-front investments in the application license, servers, people and other resources. The provider either owns the software or has a contractual agreement with the software vendor to license access to the software as part of the offering. Generally under such arrangements, the SaaS provider and has responsibility for the software upgrades and maintenance.
- *Centrally managed:* the application service is managed from a central location rather than at customers' sites.
- *One-to-many service:* SaaS tends to be designed to be a one-to-many offering.
- *Delivers on the contract:* there are many partners working together to provide a SaaS solution. The provider is the firm that is responsible, in the customer's eyes, for delivering on the customer contract for provision of the application service.

A particular form of the SaaS model is the ASP (application service provider) model, a term said to have been coined by the market intelligence firm IDC.[14] In the ASP model, the customer pays the software vendor for the application software license, while the service provider hosts and manages that application on its servers; the application is then accessed by the user through direct connections or via the Internet.

There are numerous functions offered by SaaS providers including, for example, customer relationship management (CRM), human capital management (HCM) and enterprise resource planning (ERP):

- *CRM functions* include the design and control of the pre-sale and post-sale activities of a company in relation to their customers. They include various aspects of dealing with customers such as marketing, call centre interaction, sales and technical support.[15] Providers aim to deliver improved integration within and across these areas.
- *HCM functions* aim to align business and individual performance goals and to consolidate multiple human resource transactional systems for efficiency and global workforce visibility (Forrester, 2006).
- *ERP functions* include software to manage the acquisition and deployment of resources across an enterprise.

In some cases, providers combine their SaaS role with other offerings such as *integration services* (*e.g.* business and supply-chain integration). Gartner (2006) found that "…the majority of integration service providers will offer some form of business activity monitoring (BAM) capability (for example, giving users visibility into the execution of the order-to-pay process, linking payments to their associated invoices and to their associated purchase orders, and allowing users to generate alerts on payments). In addition to expanded supply chain integration capabilities, many providers of hosted integration services also take the role of application service providers, with some form of hosted applications."

Embedded software

Embedded software generally resides on a long-term basis in hardware units other than computers, where it is used to control various product components. In contrast to software operating, for example, on a standard desktop computer, such software is almost never directly manipulated by a user, though user input may be required to specify actions or select options among the various functionalities. That is, such software is usually self-contained and not subject to user modification. Consequently, it must be "extremely reliable, very efficient and compact, and precise in its handling of the rapid and unpredictable timing of inputs and outputs".[16] Embedded software is on the way to becoming ubiquitous in modern economies, being found in a very broad range of electronic products and systems. Examples of industries with particularly heavy use of embedded applications include the automobile industry, mobile phones, robotics, tele-communication systems, medical devices and consumer electronics.

Embedded software differs from other types of software in that it often does not appear on product invoices (van Genuchten, 2007). The reason is that this type of software is generally sold as an integral part of the hardware products where it resides. However, there are some indications that embedded software developers are seeking to improve their business models by finding new ways to market, reuse or repurpose their embedded software and thereby generate direct revenues from it (discussed in more detail in Chapter 3). Recent studies indicate that the embedded software market is rapidly growing and coming to play an important role outside of the traditional software sector. Van Genuchten (2007) points to its importance for the electronics sector and foreshadows its future expansion in other sectors. A more quantitative insight is presented by METI (2007), through an enterprise survey on embedded software.[17] The results indicate that the market for embedded software development in Japan grew during 2005-2006, for example, at an annual yearly rate of 7.3% attaining a level of JPY 3.51 trillion rapidly expanding in usage across various sectors of the economy.

However, as the embedded software market expands, many non-software firms are faced with a significant challenge with respect to recruiting and training of high-quality software engineers. Many companies have difficulty in finding specialists who can bridge software and hardware. For example, automotive part manufacturing company Bosch is currently short of 2 000 software engineers. Industries are tackling the shortage of human capital by recruiting from other countries and offering systematic training. In Japan, embedded software technology skill standards (ETSS) were published by the Software Engineering Centre (SEC) of Japan with the aim of facilitating the training of highly skilled personnel for advanced embedded software development (METI, 2007). The Japan Automotive Software Platform Initiative (Jaspar) has applied the advanced software engineering methodologies (such as ETSS, etc.) to the software development project, with assistance provided by the SEC.

Generally, the market demand for software appears to be growing quite rapidly. This is perhaps not surprising, given the rise in the number and complexity of individual systems that employ software in an expanding range of applications. The growth is fuelled by such factors as globalisation, technological progress (especially in information and communication technologies, ICT) and deregulation (*e.g.* expanding the availability of radio spectrum for commercial use) and is sustained through the application of innovations and entrepreneurship to solve specific problems. Despite concrete indications of growth in the software sector, precise assessment of software production and distribution is especially challenging in comparison to markets for more traditional products (Box 1.3).

Definitions of the term "software sector" may vary in the scope of the activities that they cover. In this study, the term "software sector" is defined loosely to include the traditional "software industry" (*i.e.* companies or institutions that primarily deal with development of software such as Microsoft, Adobe or Mozilla), as well as the parts of other industries that are involved in software development. Thus, the perspective employed here includes elements of such industries as finance, electronics and government, among others, as well as some academic or non-profit institutions.

Box 1.3. The challenge of software measurement

Clear and reliable statistics are an important foundation for economic assessment of any industry. Yet, in the case of the software sector, measurement (*e.g.* determination of actual economic value) and analysis of the data are not easy tasks. There are several main reasons for this:

- *Software is an intangible good*. A given software product often reflects the culmination of a very large and costly research and development effort (see Chapter 2). Yet, as a digital product, software features an extremely low cost of duplication. Thus, marginal production costs may be a poor starting point for an exercise aimed at determining the economic value of an additional unit of software.
- *Non-traditional approaches to development, production and distribution*. A significant portion of software development, production and distribution may not be separately identified in enterprise costs and revenues (being bundled with other activity), or may occur outside of the standard commercial environment. There may be no monetary payment (*e.g.* wages or price) tracked separately for use in directly valuing the software. That is, the software costs or revenues may be embedded in other aspects of the operation (*e.g.* revenue may come indirectly such as through advertisements delivered via the software or from services provided for software support and maintenance). In other cases, development may advance partly through non-monetary incentives or external collaboration. Thus, for some software products there are no readily available proxies for use in economic valuation.
- *Variability in treatment of software statistics*. Even for the types of software (*e.g.* packaged software) or dimensions of software (*e.g.* investment in software acquisition) that are most readily tracked, there is some variation among businesses and nations in the treatment of the statistical measures. For example, accounting standards may differ in their disclosure requirements for businesses or the handling of capitalisation of software investment in national accounts may vary.
- *International comparability of price levels*. Controlling for comparative price levels across countries is a very difficult task, particularly given the intangible nature of software, a rapidly changing market environment, and the lack of observable market price for some types of software. As Ahmad (2003) points out, inconsistencies in price indices may be a significant factor in international comparisons.
- Data on software investment may be subject to particular biases, due to the effects of the accounting treatment of software investment and valuation approaches in investment surveys (Ahmad, 2003).[18]
- *Accounting treatment of software investment may introduce downward biases*. In the United States, for example, most of the software products are capitalised directly, whereas in the United Kingdom most of the software is capitalised indirectly (*i.e.* when purchased as a bundle with PCs), and hence the investment is recorded as investment in hardware, not software. In another example, French statistics report a high proportion of software expenditure as "software consultancy", whereas other countries may classify such expenditure as "capital investments". Ahmad (2003) also points out that tax schemes can create incentives for firms to underestimate the value of in-house software.
- *Investment surveys vary in their approach to valuation of in-house software.* Generally, these estimates value such software based on the estimated costs of software development (*e.g.* wage costs). However, there is variation across countries in the estimation techniques. For example, there is no consistent statistical treatment of wages and salaries across countries (*e.g.* which may – or may not – include employers' social contributions). Moreover, some countries do not capitalise in-house software if it is used for further development of other software products. Evidence from one OECD study suggests that many firms do not capitalise in-house (own-account) production of software at all (OECD, 2001).

OECD statistical measures, where they have covered specific types of software, have tended to track three main categories, whereby the first two represent software that is purchased from external suppliers (Ahmad, 2003; Lequiller *et al.*, 2003):

- *Packaged software:* software that can be directly acquired "off the shelf" from a retail store or via the Internet,
- *Custom software:* software prepared specifically for a customer by an external supplier, and
- *In-house software:* software that is developed within a company or organisation to suit its own needs (*e.g.* for use embedded in an electronic product produced by the firm).

Statistics based on these categories often omit certain segments of the software sector (*e.g.* in relation to some open source software) due to an approach that only covers software that has a price or software whose cost of creation or acquisition can be established based on accountancy techniques. In some specific cases, the omitted coverage may be significant. For this reason, the following section – which draws on available data on investment and revenue in order to substantiate the scale of the sector in the economies of OECD countries – likely understates its true scale.

Software investment

Ahmad (2003) presents a cross-country comparison of estimates of software sector investment with respect to purchased and in-house software (Table 1.2). The results indicate that, at the time of his analysis, investment in these types of software was already equivalent to sizeable shares of gross domestic product (GDP) in the selected OECD countries, equating to between 0.5% and 2.7% of GDP. In terms of GDP shares, Sweden had a sizeable lead over the other countries shown in the table, with the United States placing second. With respect to the specific types of software, Sweden led in terms of investment in purchased software and tied with Canada in terms of investment in in-house software.

The OECD Productivity Database presents estimates of the value of software investments for OECD member countries drawing on data collected by national statistical offices. The evolution of investment in software relative to total non-residential gross fixed capital formation from 1995 is shown in Figure 1.2.[19] Overall, investment in software demonstrated substantial increases for most OECD member countries for which data were available, with some variation by year and country. Figure 1.3 points that in most countries a majority of ICT investments were made in software. These figures are all the more impressive, as investment in software may still be underestimated (Box 1.3).

Table 1.2. Software investment as percentage of GDP

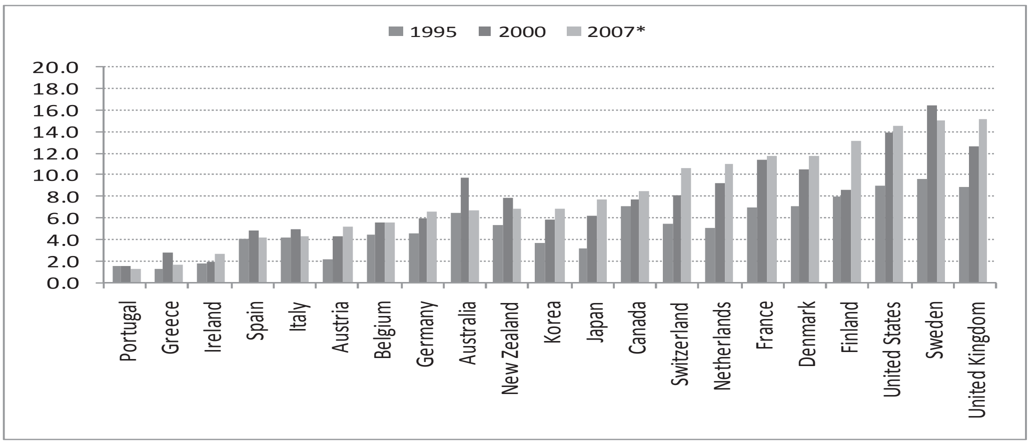
	Purchased software	In-house software	Total
Australia	1.1	0.7	1.8
Canada	0.8	1.0	1.8
Czech Republic	1.0	n.a	n.a
Denmark	0.8	0.4	1.3
Finland	n.a	0.4	n.a
France	0.6	0.8	1.5
Greece	0.3	0.2	0.5
Italy	0.5	0.4	0.9
Japan	1.3	0.6	1.9
Netherlands	0.9	0.9	1.8
Spain	0.6	0.3	0.9
Sweden	1.7	1.0	2.7
United Kingdom	0.8	0.8	1.7
United States	1.2	0.9	2.1

Notes: n.a. = not available. Based on national data, adjusted to improve comparability.

Source: Ahmad (2003), "Measuring Investments in Software", OECD Directorate for Science, Technology and Industry Working Paper 2003/6, OECD, Paris, available at *www.oecd.org/sti/working-papers*.

Figure 1.2. Share of ICT investment in total non-residential gross fixed capital formation

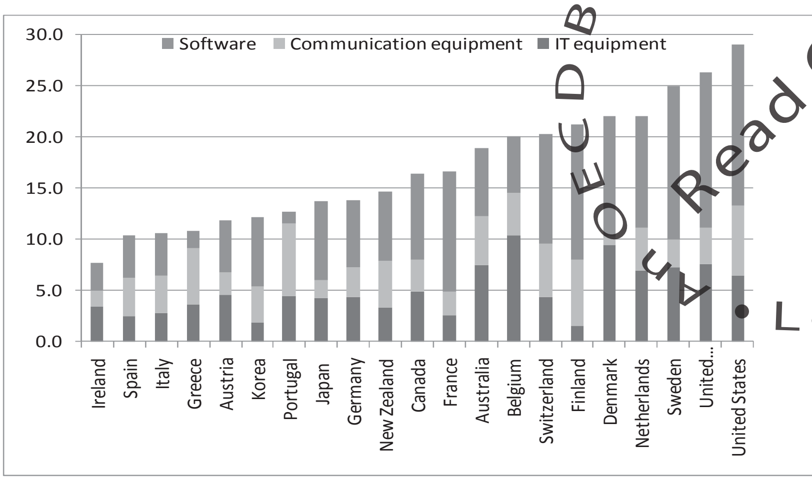
Average annual change, percentage



* or latest available year.

Source: OECD Productivity Database.

Figure 1.3. Share of ICT investment in total non residential GFCF, 2007*



* or latest available year.

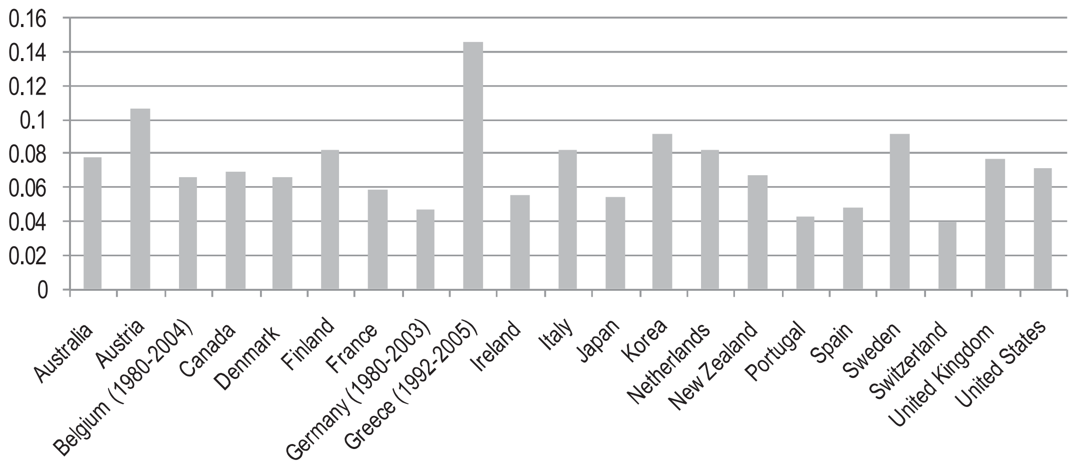
Source: OECD Productivity Database.

Software: contribution to dynamics of the economy

The growing importance of software in OECD economies is underscored by its increasing contribution to capital services. Capital services represent the flow of productive services provided by an asset that is employed in production.[20] Based on data from the OECD Productivity Database, the contribution of the software industry to the growth of capital services can be calculated for the period 1980-2004. During this time, software's contribution rose continuously in all OECD countries for which data were available, particularly the United States and Sweden (Figure 1.4).

Figure 1.4. Software's contribution to growth of capital services, 1980-2005

Average annual change, percentage

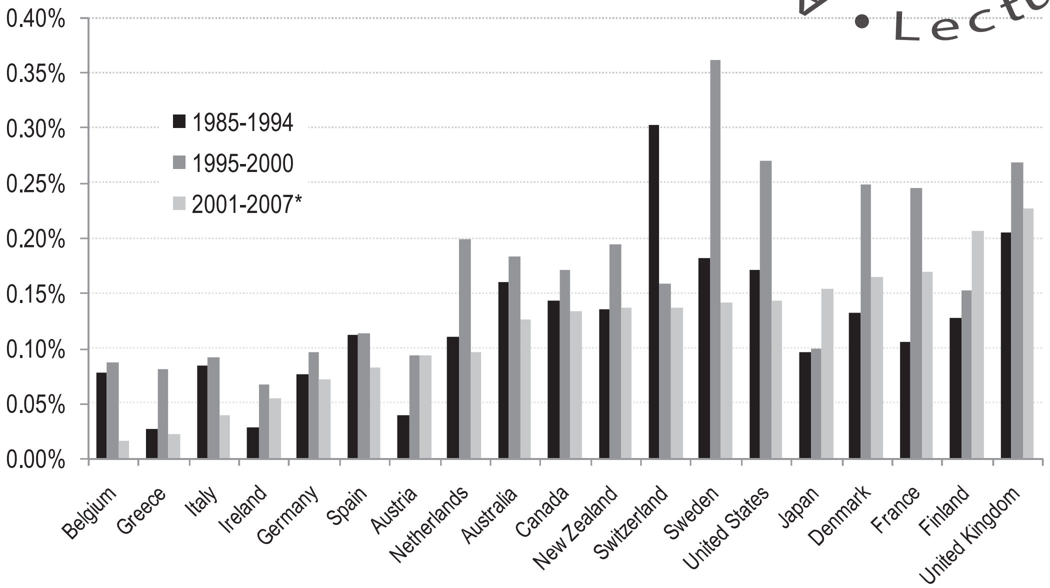


Source: OECD Productivity Database.

The estimates of capital services can be used to calculate the contribution of software investment to GDP growth (Figure 1.5). This was positive in all of the countries and time periods for which data were available. For a number of the countries, the second half of the 1990s was a time of particular strength in this indicator, with Sweden, the United States and the United Kingdom each experiencing a contribution of software investment to GDP growth greater than 0.25% per year. The chart also reveals significant period-to-period variability in the indicator in some countries.

Figure 1.5. Contribution of software investment to GDP growth

Average annual growth rates, percentage



* or latest available year.

Source: OECD Productivity Database.

The dynamism of the software sector is also reflected in its contribution to added value (*i.e.* the contribution of software as a factor of production in augmenting the value of a final product). According to one OECD study (OECD, 2006), the software sector reported the highest growth rates of value-added among the all the ICT industries. The nominal value added in the broadly defined software sector represents around 1.5% to 3% of business sector value added in OECD countries. While the share remains relatively small, it has been increasing regularly.

The highly dynamic character of software is also confirmed by recent data on financial indicators of the top 500 software companies (Software Magazine, 2008). For example, these data point to the high growth rates of overall revenues during the period 2006 to 2007 including from activity directly related to software (Table 1.3). Looking at just the top 50 companies, average revenue increases are somewhat smaller, which implies that the smaller firms in the group contribute disproportionately to the growth. This growth comes on top of an already substantial revenue base for the leading firms (Box 1.4).

Table 1.3. Top software companies: revenue growth rates
Annual change, 2006-2007

	Top 500 firms	Top 50 firms
Corporate revenue growth rate	23.5%	16.6%
of which software and software-related services	*23.3%*	*16.2%*

Source: Software Magazine (2008), "The Software Top 500 Survey", available at *www.softwaremag.com.*

Box 1.4. Revenue of the top 500 software companies

A recent survey by Software Magazine (2008) provides some insight into the economic scale of the software sector, presenting data for the top 500 software firms (out of the many thousands in the global economy). The data indicate that in 2007 the top 500 software companies as a group reported aggregate revenues of some USD 750 billion, with over one-half attributable directly to software and software-related services.

Revenue of the top 500 software companies, 2007

	Total for this group of firms	Average firm in this group
Corporate revenue (USD billions)	750	1.5
of which software and services revenue (USD billions)	*394*	*0.8*
Share of software services in total revenue (%)	52.5	52.5

Source: Software Magazine (2008), "The Software Top 500 Survey", available at *www.softwaremag.com*

Further evidence of the dynamism and innovative character of the software sector comes from White *et al.* (2004) who consider developments in software prices. The authors analysed the changes of software prices during the period 1984 to 2000[21] and concluded that that the average annual growth rates of quality-adjusted prices of personal computer operating systems were negative, ranging from -15% to -18%. Similarly, those for productivity suites generally ranged between -13% and -16%. Moreover, the price declines were generally greater in the latter half of the samples. (It may be that productivity-boosting innovation in the sector provides one explanation to reconcile such dramatic declines in prices of software with the strong data on the profitability for the sector.)

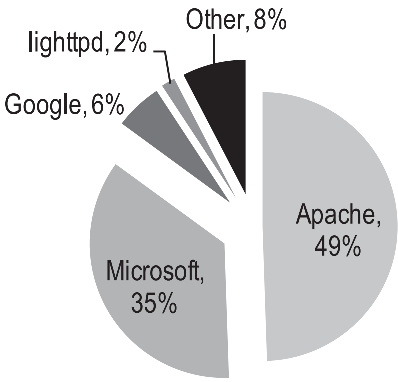
Indicators based on unit counts

A general problem with valuation approaches to software statistics is that they may fail to take into account certain types of software investment or use (as discussed Box 1.3). Therefore, as a complement to valuation-based indicators, analysts sometimes employ unit count approaches to monitor software developments. These approaches may rely on surveys or distribution data to track the distribution, installation or use of various types of software. In some cases, they rely on automated Interned-based methods. Such approaches are useful, for example, in providing information on activity related to some types of software distributed via non-traditional channels and that might not be reflected in standard economic statistics on the sector (*e.g.* software that is distributed free of direct charges to the user).[22] Although not completely free from biases, they provide a useful additional set of information for use in assessing the size and evolution of various software markets.[23]

Market penetration rates provide one example of an indicator based on such approaches. This refers to the proportion of a total available market that is serviced by a given product or product type. In the case of software, market penetration rates can be presented as the ratio of the number of users or installed units of a specific software product or product category to the total size of the market. There are no comprehensive studies that address the issue of global market penetration rates. Rather, the existing studies generally assess market penetration rates of particular software products in selected countries or selected types of software globally.

The Netcraft Web Server Survey[24] provides an example of a unit-count approach based on an automated Internet-based survey. It has tracked the growth in the number of Internet sites, highlighting the increase since 1995. The survey identifies the software used on the various sites in order to provide up-to-date information on market penetration rates (Figure 1.6). The unit counts underscore the large size of the market for the leading server software products (*e.g.* 87 million servers using Apache, 62 million using Microsoft IIS). (A review of studies on OSS market penetration rates is presented in Annex A, covering both survey and automated Internet-based exploration methodologies.)

Figure 1.6. Top developers, web servers, shares based on unit counts (in %), July 2008



Note: Shares based on unit counts from 175 480 931 identified sites.

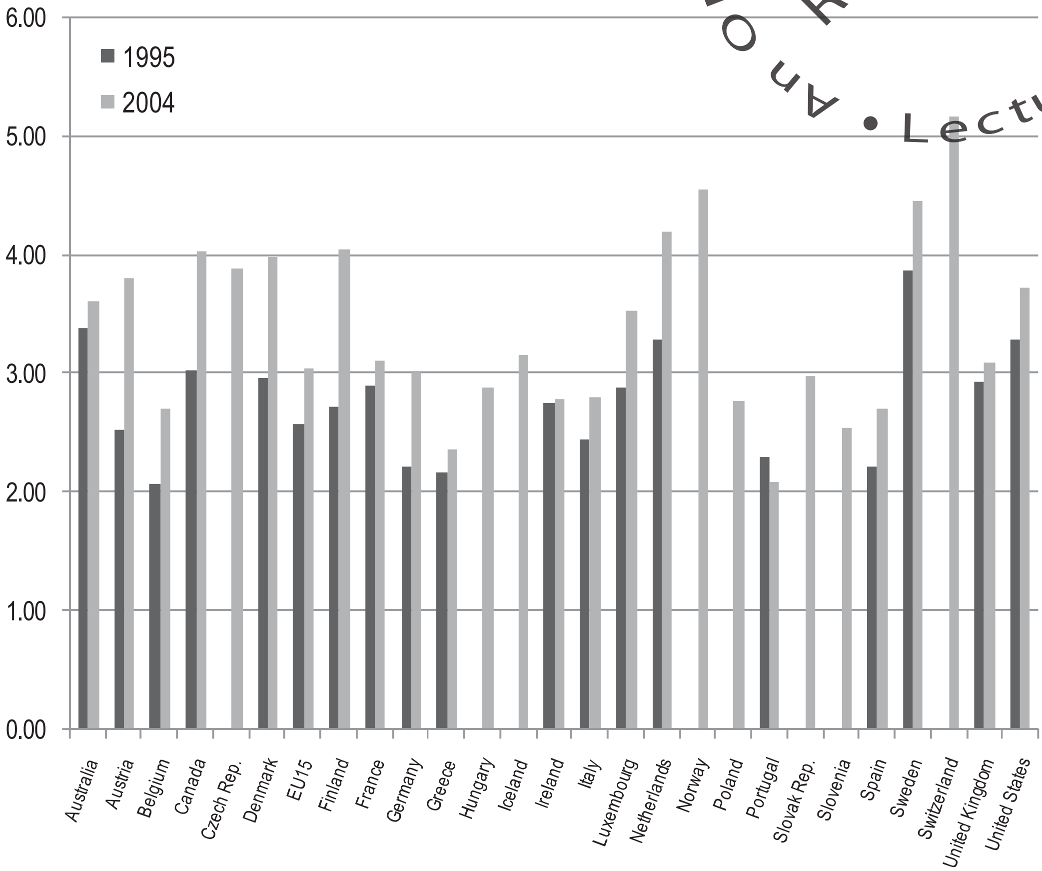
Source: Netcraft (2008), Web Server Survey, latest version available at *http://news.netcraft.com/archives/web_server_survey.html.*

Software and the labour market

ICTs, including software, have become a major source of employment creation across the OECD countries as well as a number of leading developing countries. Moreover, in view of the nature of software as an intangible product based on intellectual processes, the sector is particularly dependent on a highly skilled labour force. According to UNCTAD (2002), “The computer software and services industry is a key example of knowledge production, as the value of what a software company produces is almost entirely in the knowledge embodied in its products and services. It is a fast growing industry producing high value services for its customers.”

Employment in the ICT sector has experienced growth in most OECD countries for the last decade. Figure 1.7 presents data from 27 OECD countries on employment in the ICT sector in 1995 and 2004. In all the countries except Portugal, the share of ICT specialists in total employment increased during this period, in some cases substantially. As of 2004, six of the countries reported ICT-related employment shares of greater than 4%.

Figure 1.7. Share of ICT-related employment in total employment, 1995 and 2004[1,2,3]



1. ICT specialists, who have the ability to develop, operate and maintain ICT systems. ICTs constitute the main part of their job.

2. In some cases, the data may not be strictly comparable: *i)* across time due to changes in definition and *ii)* across countries, as data classifications for EC and non-EC member countries have not been harmonised.

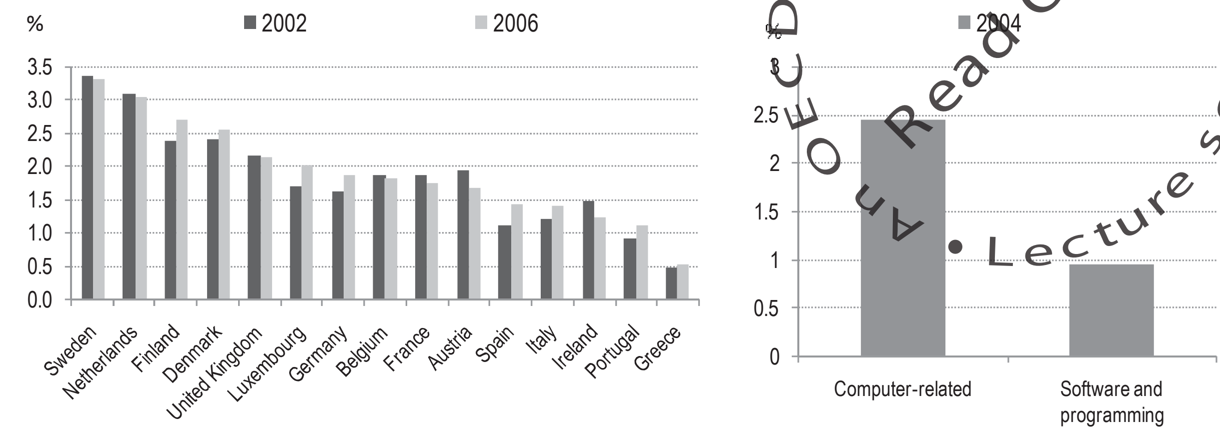
3. Data are for 1995 and 2004, except: Australia, Finland and Sweden 1997 instead of 1995; Portugal 1998 instead of 1995; Ireland 1999 instead of 1995; Austria and Canada 2003 instead of 2004.

Source: OECD (2006), *OECD Information Technology Outlook 2006*, based on EULFS, US Current Population Survey, Statistics Canada, Australian Bureau of Statistics.

More specifically, and for a more recent period, data on the share of computer-related employment in total employment are available for the EU15 and the United States. This indicator ranges from about 0.5% to 3.5% for the EU countries during 2002 and 2006 (Figure 1.8).[25] For the United States, a similar measure corresponded to around 2.4% of total employment between in 2004. As the US data are more detailed, software specific occupations can also be identified; the data show that computer programmers and computer software engineers accounted for around 1% of total employment in the same year.

Figure 1.8. The share of computer-related employment in total employment

EU15, 2002 and 2006 (left)
United States, 2004 (right)



Note: For the EU countries, the ISCO88 categories "213 computing professionals" and "312 computer associate professionals" were used as a proxy. For the US, the CPS categories "110 computer and information systems managers", "1000 computer scientists and systems analysts", "1010 computer programmers", "1020 computer software engineers", "1040 computer support specialists", "1060 database administrators", "1100 network and computer systems administrators", and "1110 network systems and data communications analysts" are used.

Source: Authors' calculations based on EULFS and US Current Population Survey (CPS).

Overall there has been a tendency for growth in ICT employment, albeit with some variation by country in recent years. This has been accompanied by increased demand for highly skilled workers to meet the demands of new and changing trends for particular technical skill sets. For example, in the United States as of 2006, the Bureau of Labor Statistics (BLS, 2008) estimated employment in just one software related category – computer software engineers – to be about 860 000. Through 2016, BLS projects this number may increase by 38%.

The demand for highly skilled labour has its origin in the nature of software innovation, which is not a physical capital-intensive activity but rather a creative human capital-intensive one. A software developer needs to be able to understand various aspects of a given (often complex) task and to address them through software modelling. In doing so, a software developer needs to display the ability to deal simultaneously with numerous details of a software task and, and to possess well-developed abstract thinking skills. Software developers work mostly in teams. Software teams display high levels of team learning, constructive software practice patterns, excellent communication, appropriate role division and good integration of expertise. Even though some software teams are associated with project work in conventional software development firms, they may also be geographically distributed and supported by co-operative work systems. They can also be self-organising, facilitated by the Internet, and motivated by non-commercial goals, as in open source communities. The importance of human capital input in the software sector was underscored in replies to the OECD business questionnaire (Box 1.1 and Table 1.5), where trained human capital was selected as the most important feature for successful innovation activities (OECD, 2008a).

In view of the large and growing employment shares related to software, OECD countries are being challenged to educate increasing numbers of graduates with skills appropriate for employment in the ICT area.[26] Enterprises as well are being challenged in recruiting and retaining talent, which is a very important issue for firms looking to remain competitive in terms of innovative software development.[27] Clearly, the human capital dimension is set to play an important role in relation to software innovation.

Software innovation: key features

Concept of software innovation

Innovation may be broadly defined as the successful commercial introduction of a new product or process. More specifically, according to the OECD's *Oslo Manual* (OECD, 2005), innovation refers to "implemented technologically-new products and processes and significant technological improvements in products and processes. An innovation has been implemented if it has been introduced on the market (product innovation) or used within a production process (process innovation). Innovation involves a series of scientific, technological, organisational, financial and commercial activities." In this context, it is important to note that research and development (R&D) activity is just one element – albeit an important one – in the process of software innovation.

Software innovation can be seen as a process leading to:

- development of a novel aspect, feature or application of an existing software product or process; or
- introduction of a new software product or process or an improvement in the previous generation of the software product or process; and
- entry to the market or use within the production process.

Box 1.5 provides an illustrative list of innovations on the horizon in software.

Incentives are the fundament for purposeful activities including those that lead to software innovation. Direct economic incentives can be provided through earning economic rents, and there are various sources of rent for an innovative software developer. Most of them refer to the profits earned, but can also include lower costs of production or increased efficiency of production and service delivery. Some of the economic rents are directly associated with the product (*e.g.* profits earned through sale of the software product or through sales of hardware with embedded software), others are linked to revenues from activities, such as services or advertisements that are provided by the software developer together with software product. With the advent of such developments as cloud computing and collaborative approaches, the possibilities for finding revenue "sweet spots" are becoming more diverse.

Beyond the direct monetary incentives, there are also quasi-non-monetary incentives for software innovation.[28] While these may not yield immediate economic returns, they often offer some long-term economic advantage. In the case of an enterprise, examples include enhancing the firm's reputation or market share, testing a new product concept, or exploration of new markets. In the case of individuals, this may include gaining experience and education, fame, signalling of skills or ideological motives (*e.g.* community identification or altruism), among others.

Box 1.5. Examples of software innovation on the horizon?

The software press and Internet media offer hints of a wide range of innovation just over the horizon. A few illustrative examples are listed below.

Business applications

The Enterprise 2.0 Conference (*www.enterprise2conf.com/*) sees innovation as emerging to deliver a set of "technologies and business practices that liberate the workforce from the constraints of legacy communication and productivity tools like email. It provides business managers with access to the right information at the right time through a web of inter-connected applications, services and devices. Enterprise 2.0 makes accessible the collective intelligence of many, translating to a huge competitive advantage in the form of increased innovation, productivity and agility."

User interface

Craig Mundie, Chief Research and Strategy Officer for Microsoft Corporation, argues that one of the next big innovations in computing will involve user interface and accessibility (BBC Interview, 16 May 2008, *www.bbc.co.uk/worldservice/specials/924_interview_archiv/page12.shtml*). He foresees change in the next five to ten years allowing the computer to interact with users more in the manner of a human being, "to allow many more people to have facile use of computing" through language advancement (*e.g.* spoken commands, use of multiple languages and increased computer vision). He notes that some on board computers in cars use spoken-command inputs, constituting an early example of this type of innovation.

Network-user interface enhancement

Further evolution of these interfaces may result in more standardisation, with the Internet and operating systems more closely integrated. As one source puts it, "All the major developers of operating systems for PCs, network computers, and workstations, including Apple, Microsoft, IBM, and Sun, are hard at work integrating […]."
(EMC Publishing, *www.emcp.com/intro_pc/reading4.html*).

Enterprise application integration (EAI)

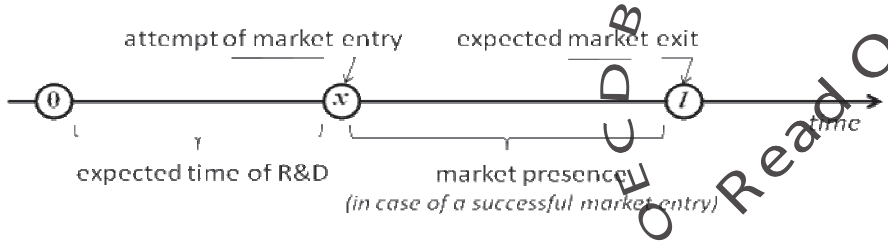
Gartner (2001) sees EAI as "unrestricted sharing of data and business processes among any connected application or data sources in the enterprise." Enterprise networks will permit better connection of applications, allowing for linkages between internal IT departments, distributors, suppliers, subsidiaries, partners, and clients. According to Wintergreen Research (Wintergreen, 2006), companies that achieve faster time to market through EAI can achieve competitive advantage.

Patterns of software innovation

Basic analytical framework

The microeconomic literature often employs a simple, basic framework as a starting point for systematic analysis of innovation mechanisms in any sector. This framework conceptualises the innovation process as a linear sequence of events; it starts from the premise that, innovators can be viewed as economically rational agents, motivated by a specific set of goals.[29] In striving to attain those goals the innovators undertake a process beginning with their recognition of a potential innovation, followed by scoping of the idea, research, prototyping, testing, introduction of the new product to the market and full commercialisation (Figure 1.9).

Figure 1.9. Software innovation: stylised sequence of events



At the R&D stage, a potential innovator launches purposeful R&D activity in order to create a product with particular goals in mind. Research time is costly either in monetary terms (*e.g.* wages, monetary cost of investments) or in non-monetary terms (*e.g.* time and effort devoted to R&D). Some of the determinants that affect the cost of research include the availability of human capital, the stock of public knowledge (*e.g.* resulting from research conducted in the public sector or from strong business-academic links), access to foreign inventions and to previous developments in the industry, and a sound financial market together with availability of direct public financial support for business R&D[30] (Jaumotte and Pain, 2005a, b, c, d).

In the economy generally, innovation tends to be associated with purposeful R&D activity whose intensity depends in turn on economy-wide conditions and specific science-related policies and institutions (Fagerberg *et al.*, 2004, Jaumotte and Pain, 2005a, b, c, d). However, specific cases of innovation may not have a proportionate relationship to research; efforts may focus on expansion and recombination of existing findings rather than new discoveries. In some cases, innovation (in the sense of successful commercial introduction of a new product or process) may take place without requiring extensive R&D.[31]

In the case of software, the process of R&D can either start from scratch or build on previous innovations, often involving some degree of collaboration. R&D in the sector is often cumulative, with existing programming “solutions” forming the starting point for further developments. Thus, software development often takes the form of a multi-stage process that builds on previous innovative steps. As a result, releases of new products and product enhancements may take place relatively frequently.

A product that is ready for commercialisation moves into the *market entry* phase. The outcome of this phase is uncertain. Products may fail to navigate the various barriers to entry and competitive pressures. Naturally, this is a critical moment in the innovation process. Commercialised innovation is in many ways *the successful exploitation of new ideas.* In a competitive market environment, when the R&D process delivers the first-best solution to a consumer need, it may seem logical that successful commercialisation should follow. A rational consumer might be assumed to choose the highest quality product *cæteris paribus*. Thus, in the case of cumulative innovation in the software sector, a significant product enhancement might tend to be followed by successful introduction to the market. However, the existing economic literature points to several known general market features that could influence the likelihood of a successful commercialisation of new innovations. There are at least three main groups of such features including product quality, the degree of competition and network effects.[32] *Market presence* is critical to attainment of the objectives of most software developers. Generally, there are two types of developer objectives related to the incentives for innovation: monetary (maximisation of monetary profits during market presence) and quasi-non monetary.

As for *market exit*, a long expected product life extends the duration of the period over which the objectives of a developer are achieved and sustained. However, compared to products in other industries, the length of life cycles for specific versions of software products may be rather short (FTC, 2003). While this depends in part on displacement by new generations of software (Harter *et al.* 2000), it is also affected by the quality of IPR protection (Jaumotte and Pain, 2005a). Weak legal standards or poor IPR enforcement result in high rates of digital piracy that can, in turn, significantly reduce potential profits for software developers (BSA, 2007b). As Feller *et al.* (2005) note, a sound IPR regime is a necessary protection for innovators from unwanted expropriation of their creations by third parties.

This conceptualisation presents a simplification of reality in the software sector. In practice, software developers may engage in collaborative approaches at any point in the process (discussed in the next section) and may employ a variety of business models to achieve objectives or extract value. In some cases, for example, developers may focus on their comparative advantage in delivering just a part of the process (*e.g.* contributing testing services or marketing and distribution expertise in a collaborative project); in others they may take the lead throughout a project. Larger firms may employ different approaches in different projects that are running simultaneously.

Collaboration and open innovation[33]

Collaborative and "*open*" approaches may be used to enhance the basic innovation process in cases when a software developer wishes to leverage innovation efforts by bringing external knowledge or capacities into the software development process. Generally there are three pillars of open innovation: *i*) sourcing and integrating external knowledge from customers, suppliers, universities and research organisations, or even competitors, *ii*) bringing ideas to market (not necessarily finished products); selling, licensing, trading intellectual property as part of strategies to multiply technologies; and *iii*) working in alliances in order to capitalise on complementarities (Chesbrough, 2003; Enkel and Gassmann, 2004; Chesbrough *et al.,* 2006; OECD, 2008c).

Although collaboration in software innovation initiatives takes a variety of forms, some common examples might include:

- engagement of individual PhD and postdoctoral researchers in innovation projects;
- partnerships or contract research with academia through framework agreements, often with a focus on long-term co-operation (including the creation of joint laboratories or high-tech zones by ICT firms (including SMEs) on university campuses);[34]
- industrial technology alliances, R&D partnerships and consortia between ICT firms (*e.g.* for upstream research or for joint product development) or between ICT firms and partners or customers outside the sector; and
- prospecting for valuable new ideas from individuals, software communities and start-ups with promising research (including through venture capital, incubation and acquisitions).

The software industry works actively to capitalise on such external sources of knowledge. Major incentives include cost and risk reduction (especially for pre-competitive R&D) and improved possibilities to enter markets with jointly developed technologies (Freeman and Soete, 2007).[35] These collaborations are increasingly international, spanning various segments of the ICT sector and adjacent industries (*e.g.* biotechnology).

The use of collaborative and open approaches has been strengthened by globalisation and technological progress. Internationalisation is facilitated by the increasing use of ICTs as a basis for the science and technology infrastructure (*e.g.* broadband-linked research networks) and by the fact that international research collaborations have been encouraged by policy programmes (*e.g.* the EU FP7's focus on research co-operation with entities from Asian countries) and specialised organisations (*e.g.* the International Technology Roadmap for Semiconductors).[36] Long-standing public research organisations with dedicated ICT foci (Fraunhofer in Germany, Battelle in the US, VTT in Finland, TNO in the Netherlands) are also increasingly engaging in global research alliances in order to leverage their efforts. Moreover, economic liberalisation has facilitated the establishment of international business ties among enterprises via trade and investment.

Software innovation: beyond the software sector

Software development takes place across the economy. The tight links and complementarity of the software sector with other industries are highlighted in several OECD studies (OECD, 2002, 2004, 2006a, 2006b). These studies point to the relationship of the software sector to information, computer and communications industries through technological links and through the business activities of firms that are often present in multiple markets.[37] Moreover, a substantial volume of software is produced outside the traditionally defined "software industry" (*e.g.* van Genuchten, 2008), as is often the case for software developed in-house or for products in sectors such as consumer electronics or cars.

Data on R&D expenditures related to software underscore the breadth of such investment across the economy, a point confirmed in several national R&D surveys.[38] According to Young (1996), for example, at the beginning of the 1990s, about 40% of services firms in Japan and Italy undertook some form of IT research activities including software development. In Denmark, about 75% of all R&D investment reported by "other services industries" was computer-related. In Canada, over half of all R&D in the services industries was software-related.

In-house needs constitute a major driver of software development in firms beyond the sector. Even though the exact quantification of software remains a major challenge, existing estimates suggest that software developed for own needs ranges from 20% to 40% of software production. A further 40% to 50% of the market is in custom software, produced for users by software service providers (Parker and Grimm, 2000; Grimm *et al*., 2002).

Technical aspects of software innovation

Software innovation occurs in an abstract and digital space, which makes it technically different from some other industrial disciplines such as electrical or mechanical engineering. Whereas other areas may be subject to numerous physical constraints, software development, by contrast, is physically constrained only by hardware for which it is designed. Within this relatively unconstrained environment, the software development process strives to optimise across a number of dimensions in what can be a highly complex task.

Development of a high quality software product may require simultaneous consideration of such characteristics as functionality, reliability, usability, efficiency, maintainability and portability, among others (ISO, 2003). The software sector has responded to this complexity in two general ways. On the one hand, there has been an effort to advance the field in a rigorous manner based on engineering disciplines. On the other hand, software innovators have aimed to create certain frameworks that enable less advanced users to apply their ideas in developing software to meet their specific needs. These two concepts are discussed below.

Software engineering and development tools

Software engineering aims to apply engineering approaches to software development, deployment and maintenance. It is a relatively new field, with the term "software engineering" first established during a conference of the NATO Science Committee held in 1968 (NATO, 1969). SWEBOK, the IEEE (2004) Computer Society guide to the field, defines software engineering in terms of "the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software…".[39] The guide lists ten knowledge areas covered by the field, including: software requirements[40], design, construction, testing, maintenance, configuration management, and quality, plus software engineering management, process, and tools and methods.

Software engineering tools consist of programs or applications that assist in creating, configuring, maintaining, testing or supporting software-intensive systems. These tools evolve continually in response to the development of new methodologies for creating software systems. Over the years, they have become quite sophisticated and now cover the full lifecycle of software systems. They are often bundled together to create an integrated development environment (IDE), offering access to toolkits for software development in one or more programming languages. Illustrative examples of IDEs include *Eclipse* (an open source solution), JetBrains' IntelliJ IDEA (a Java IDE) and Microsoft Visual Studio (a Windows IDE).

Progress in software engineering and development tools has been driven or facilitated by a number of factors including, for example:

- *The exponentially growing demand for software systems, which has increased in terms of both the number of systems and the complexity of the individual systems.* Moreover, software development is very skill-intensive and labour-intensive. Consequently, participants in the software sector have recognised that new tools, methodologies and processes are required to enhance their productivity, satisfy quality requirements and fulfil demand.[41]

- *Advancement in hardware technologies, including processor speed[42], display technology, storage capability and networking* (Taylor and Van der Hoek, 2007). For example, increases in processor speed have boosted the analytical potential of tools and high-speed network communications have broadened the participation of stakeholders in the software design process.

- *The need for a degree of consistency in a changing environment and the push for technological convergence.* The extremely complicated, heterogeneous and dynamic situation in the software sector is challenging software developers to take into account of multiple environmental aspects and contexts while delivering consistency of software design decisions and multidimensional access to design data (Taylor and Van der Hoek, 2007). Advanced approaches and tools are needed to tackle some of these challenges.

- *Globalisation, enhanced power and diffusion of ICT infrastructure, and the tools supporting collaborative work* have enabled software engineers to engage expanded resources around the world for purposes of research and development.

In sum, advances in networks and computing power have resulted in an explosive growth in sophisticated applications and services that touch almost all aspects of society. Software engineering is playing a central role in this process, both contributing to these advances and helping to respond to the demand for further progress.

Software frameworks

The appearance of software frameworks during the 1990s marked a new approach to software development, particularly with respect to applications. A software framework is "the skeleton of an application that can be customised by an application developer" (Johnson, 1997).[43] Software frameworks can include support programs, code libraries, a scripting language or other software to help develop and assemble the different components of a software project. Frameworks facilitate software development by allowing designers and programmers to spend more time on delivering functionality and less time on the more tedious low level details and routine aspects of providing a working system. Thus, a high-quality framework increases developer's productivity, in particular for those involved in big and complex projects.[44] A related approach is the use of service-oriented architecture (SOA) in order to promote efficient development of software systems (Box 1.6).

Box 1.6. Service-oriented architecture (SOA)

Under service oriented architecture (SOA), enterprises and other users can structure their computer systems in modules that can be accessed in various combinations and by various users as necessary to deliver services. SOA, through its emphasis on modular design, permits inter-operation of numerous services, where access to these services can be executed autonomously. This is achieved by imposing a certain structure on the basic applications and introducing a set of smaller flexible modules that can be created and modified according to current demands (Bell, 2008). According to a report of WinterGreen Research (2006), this segment of the market is set to grow rapidly over the next few years. The authors state, "SOA markets at $450 million in 2005 are expected to reach $18.4 billion by 2012."

In the construction of frameworks, two types of elements can be distinguished (Pree and Sikora, 1997). The rigid and immobile elements ("frozen spots") of frameworks provide the necessary structure and cannot be changed by a developer; they are the fundamental and basic elements of a framework system that shapes its overall architecture. Programmers focus on the flexible elements ("hot spots") in the framework, which they can modify in order to meet a given project's functionality requirements. There are numerous fields of application of software frameworks including: graphical editors, music programs, programming software, multimedia and web applications, among others. Examples of software frameworks include Java Native Interface, Salesforce.com's Force.com, .Net Framework 3.5, and Apache Struts.[45]

Business models and software innovation

A business model might be considered as “the translation of strategic issues, such as strategic positioning and strategic goals into a conceptual model that explicitly states how the business functions. The business model serves as a building plan that allows designing and realising the business structure and systems that constitute the company’s operational and physical form” (Osterwalder *et al.*, 2005). With technologies evolving rapidly in the software sector, strategic issues continue to shift and new ways of doing business are emerging. In many cases, firms are finding it necessary to re-examine their business model and find alternative ways of raising revenue. The interaction of business model and technological developments can stimulate – or be stimulated – by innovation.

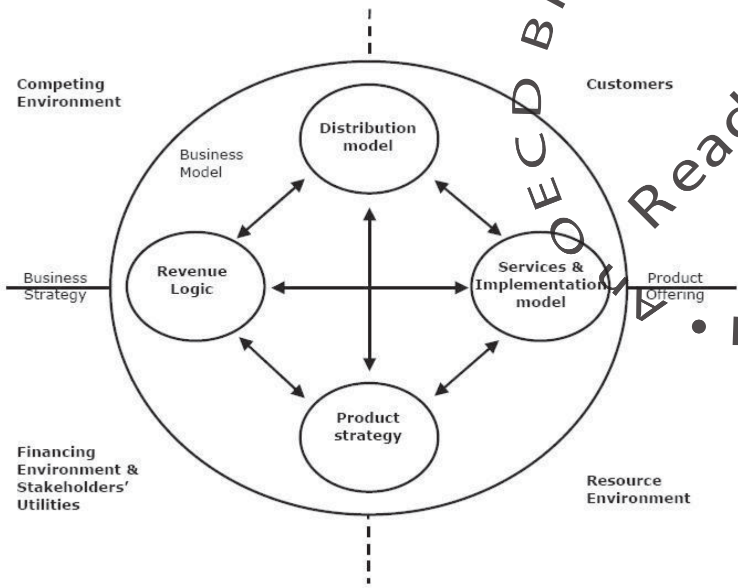
During the 1950s and 1960s, IBM and other major providers frequently employed a revenue development strategy by which their software was provided free of charge or bundled with proprietary hardware with the goal of increasing revenue from the sale of hardware (Campbell-Kelly and Garcia-Swartz, 2008). As demand for software grew in complexity, hardware producers (notably IBM) shifted to more open design approaches for hardware that enabled external software developers to deliver innovative software and as more hardware became accessible to benefit from significant economies of scale (van Genuchten, 2007).

Nowadays a broad range of business strategies and practices are employed by various commercial and non-commercial entities in the development and distribution of software products and services. No single development strategy or business model has gained primacy. Rather, the software sector is characterised by its dynamism, with a shifting range of commercial strategies and practices and a diversity of market participants. The proliferation of models is largely the result of new platforms[46] and distribution channels, in particular with ever increasing broadband connectivity.

One useful approach to analysing business models for software firms is provided by Rajala *et al.* (2003). Noting that each company and business model is constrained by external variables such as the macroeconomic framework conditions, financing conditions and the competitive environment, these authors decompose a generalised business model into four elements (Figure 1.10):

- Product strategy: describes the core product and service proposition of a software business and the way the product development work is organised.
- Revenue logic: describes the sources of revenue and the way the software vendor generates revenue from these sources.
- Distribution model: describes the way marketing and sales of the product and service offering have been organised and identifies the sellers and marketers of the product and service offering.
- Service and implementation model: describes how the product offering will be dispatched to the customers and deployed as a working solution, including the set of services and actors implementing them.

Figure 1.10. Elements of a business model



Source: Rajala *et al.* (2003), "A Framework for Analyzing Software Business Models", *http://is2.lse.ac.uk/asp/aspecis/20030126.pdf*.

Table 1.4. Alternative types of software business models

Consumer models	Business software models
Proprietary software	Open source
Public domain	Public domain
Freeware	Bundled with hardware (*e.g.* HP Openview)
Shareware	Proprietary software
Adware	Consulting ware (*e.g.* E&Y's tax readiness software; obtained when the audit contract is signed)
Demoware	By the seat
Trialware	By the named user
Hardware supported (ships with hardware)	By concurrent user
Pluginware (available if you buy the main application or OS)	By average number of users
Premiumware (basic version is free, added functions require payment)	By location
Subscription ware (*e.g.* anti-virals)	By server
Upgradeware (*e.g.* tax software - cheaper if bought during previous year)	By CPU
Permanent ware (buy once and all the future versions are free)	By CPU class
Donationware (donate to a charity and it is free)	By transaction
Bundleware (second and third apps in a bundle)	By revenue
	By employee count
	Bundleware (*e.g.* Microsoft's .NET strategy)
	Adware
	Donationware (normally to the software creator)
	Custom software
	Toolkits (Lodestar is a great example - buy the pieces and we will assemble your app)
	...and probably 1 000 more

Source: Discussion forum on *Linkedin*, post by D. Houseman, Associate Director, Capgemini, 19 May 2007.

There are many alternative ways of classifying business models, although most of these can be brought back to the four main elements described above. Table 1.4 gives one example of a list posted on a discussion forum on the theme "Trying to list all possible software business models".[47] Certainly, many business models exist, employing various approaches to each of the four elements. These approaches can be operationalised via a multitude of business policy levers concerning such issues as openness of innovation processes and strategic alliances, development of direct and indirect revenue sources, and approaches to product licensing, among many others.

A review of information on business models collected by the OECD Secretariat from public sources (Box 1.7) confirmed that individual firms employ different mixes in different contexts as they pursue innovation and growth strategies. The following examples drawn from these sources illustrate just a portion of the diversity of approaches applied within these firms and across the sector:

- CA Inc. uses so-called "flexible licensing" as part of its business strategy.
- Electronic Arts sells interactive games with advertising both in their games and on their website.
- Lawson relies on a combined approach of software licence fees, consulting, and training and implementation services.
- Microsoft applies different models in different contexts, in some cases using traditional models based on direct licensing and sales, often pursuing partnerships, sometimes aiming for indirect revenue streams (*e.g.* via embedded advertising on websites), and employing various degrees of openness, among other approaches.
- Misys specifically engages in partnerships and collaborations to innovate and capture new market opportunities.
- Hitachi and Oracle specifically engage in mergers and acquisitions.
- Sun Microsystems focuses on sales of hardware products, and simultaneously supplies software based mostly on the open-source licensing principle.
- Atari has a global strategy of local presence to adapt their products to the local market; they have specifically built up a broad-based catalogue in order not to rely on a single product; and they combine franchises and licences.
- Novell and Linux are two examples of hybrid structures, combining open innovation and other business model elements.
- Mozilla is an example of a foundation seeking to promote software innovation through an open framework.

The case of embedded software merits special attention in that the business models involving embedded software generally create value by selling the hardware product to customers, whereby the software contributes indirectly to the bottom line. For example, the software may add functionality valued by the consumer or permit more economical production for the producer (*e.g.* using a software switch instead of a hardware switch). There are also software developers that create value from such software directly, by selling embedded software products, as such, to hardware producers or integrators that incorporate the software in their products for subsequent resale as part of a hardware product.

Box 1.7. Information from public sources on software business models: data compilation method

In conducting its preliminary assessment of business models in the software sector, the Secretariat built a small database drawing on public information. The primary sources included annual reporting by listed companies and public statements by various non-profit organisations; where available, this was based on reporting for 2007. The website of the US Security and Exchange Commission (*www.sec.gov*) proved to be a valuable source, as US-listed firms are required to submit periodic reports with certain standard information; these include both domestic and international software-oriented firms listed on the New York Stock Exchange (NYSE) as presented on 10-K and 20-F Annual Report forms. For privately held companies or international firms not listed on the NYSE, corporate websites were used to attain applicable information from annual reports and investment profiles. Additionally information from the NYSE European partner website Euronext (*www.euronext.com*) was used in a few instances. In order to compile information on open source foundations, the Secretariat directly accessed official foundation websites. For firms developing embedded software, information could be obtained from both the SEC website and corporate websites. Articles pertaining to collaboration between software-oriented firms, open source foundations and firms utilising embedded software were taken from corporate and foundation websites directly. The data were compiled in a spreadsheet detailing information on business models, research and development, functionality and user input to innovation in order to draw comparisons related to their innovative activities. The exercise yielded information on innovation, interoperability, embedded software, open source, proprietary rights, collaboration, internal versus external development, and functionality, providing inputs for the subsequent analysis.

Approaches to licensing

Approaches to licensing vary across a spectrum from closed-source licensing and open-source licensing. At one end of the spectrum, closed-source licensing focuses on innovation through the use of proprietary licenses and intellectual property protection.[48] The idea is that keeping the source code closed, and relying on strong IPR or technological protection provide an incentive for software development by providing a means of appropriating monetary rewards for innovative thinking. Proprietary models keep source codes as property of the company and have tended not to allow for open-access and modification or alternative application of the code. Other closed-source models primarily employ technological protections or closed distribution models (*e.g.* software as a service) to prevent disclosure of source code.

Firms using proprietary models have a variety of channels for fresh innovative inputs. Some firms diversify by acquisition of firms that have innovated using hybrid models. User-driven development has also often been an important aspect of the innovation strategy in these business models. Moreover, firms using proprietary approaches may make available to outside developers information required for interoperability (*e.g.* API information). For platform developers, this can be essential in ensuring adoption, while for other types of software (*e.g.* application and middleware) interoperability can help to promote development of follow on innovation that adds further value to the original product. Ecosystems can develop around such software.

The other end of the spectrum is occupied by a range of open-source licensing models. The premise of these models is that innovation occurs when multiple developers are able to access source code in order to further the functionality of a program. Developers co-operate under a model of rigorous peer review and take advantage of knowledge, talent and skills globally, which has proven conducive to innovation. A firm allows users access to subject matter protected by its intellectual property (*i.e.* source code and technical documentation) and grants them some combination of the IPR rights needed to use, modify, and distribute that content. In return it receives input in the development of tools that it would otherwise have had to come up with on its own. Open source licences support this strategy in two ways (Hope, 2003): *i*) users cannot become co-developers of a tool unless they have access to that tool in a form that they can understand and modify, and *ii*) a requirement to provide access to source code in order to provide an incentive to contribute to a co-operative effort which would be hindered if potential contributors expect to be prevented from using the tool that they helped to create.[49]

In practice, many software developers employ a combination of various licensing models and mix of revenue-generating strategies. For example, Red Hat supplies proprietary software on top of its Linux offering. IBM offers integrated combinations of open source and proprietary software. Microsoft employs a wide range of business models and strategies, ranging from fully proprietary "closed source" offerings to "free ware" and fully open source offerings.[50] In some cases, open-source foundations such as the Linux and Mozilla Foundations support the business models of open-source companies, providing intellectual property strategies, quality control and financial support for open-source initiatives. Some open source initiatives receive corporate sponsorship in addition to the revenue they may generate (*e.g.* from donations and philanthropy) in the course of their operations. It is very probable that such corporate contributions are made with strategic considerations in mind (*e.g.* development or promotion of a specific technology or encouragement of competition in a key area).

Open source software can serve as the basis for a variety of business models[51] although this is sometimes misunderstood. Even though open source can be "free" to developers and users, there are a variety of businesses built around this type of software. Examples of business models utilised in this segment of software include:

- *Professional services:* customisation, support, training.
- *Dual license:* sell a commercial version of the software to businesses that want to use it for commercial purposes and give away a free version to the open source community.
- *Enhanced product:* sell a commercial version that has substantially different features than the open source version for consumers.
- *Software as service:* in some cases, give away the software code, but sell the software as a hosted web service either through advertising or subscription fees.
- *Bookware:* give away the code, but sell the documentation.
- *Marketplace:* give away the code, build a community marketplace, and take a cut of what all the people selling add-ons make.
- *Foundation:* get corporate sponsorship and give away the code. (*e.g.* Apache).
- *Hybrid:* blend several of these.

Across the spectrum of licensing approaches, collaboration is an increasingly important element of business models and innovation strategies for software firms. This collaboration can take place between firms that develop proprietary software or with open source organisations. There have been many initiatives to combine the benefits of open source and proprietary models to enhance innovation, as illustrated by the list of models presented above. The idea is always to take advantage of the resources available through open source and then add value that can generate revenue. The previous examples of collaboration are often achieved through the commercialisation of open source software and sometimes through the use of patent agreements between open source and proprietary businesses or through a combination of open source and proprietary initiatives (Box 1.8). Interoperability becomes increasingly important in the specific context of such software business models, with mergers and acquisitions, collaborations, outsourcing, and offshoring.

Box 1.8. Examples of "hybrid" models

When an open source product is sold or commercialised with substantially different features this can often enhance the product, which is an important aspect of innovation within business models in terms of appealing to the end-user. It is also possible to provide the software code as open source while commercialising specific software applications as a hosted web service either through advertising or subscription fees (software as a service). Bookware is another alternative incorporating proprietary and open source initiatives in which the code is presented as open source but the documentation is commercialised. In a "marketplace" model the code is presented as open source on which a community marketplace is built and profit is made through the sale of add-ons, to both individuals and sponsoring businesses. Dual-licensing is another way that software companies can combine the benefits of proprietary and open source initiatives through selling a commercial version of software to businesses, as well as giving a "free" version to open source communities. In this way businesses can capitalise on their innovations while also contributing to open innovation. Other examples of collaboration come in the form of acquisition or creation of hybrid business units. Hybrid business models often combine elements from each of the examples mentioned above or focus on one primarily, but in general they are used to increase innovation through some form of collaboration. This is often an option for proprietary companies who wish to commercialise open source initiatives under their own legal structure.

The environment for innovation

The intensity of innovation activities is influenced by a variety of factors. Recent OECD research has highlighted certain of these, including the availability of human capital, the stock of public knowledge (*e.g.* resulting from research conducted in the public sector or from strong business-academic links), access to foreign inventions, a sound financial market together with availability of direct public financial support for business R&D,[52] and the regime governing intellectual property rights (IPR, addressed in more detail in Box 1.9.) (Jaumotte and Pain, 2005a, b, c, d).

The results from the OECD business questionnaire (Box 1.1 and OECD, 2008a) are helpful in identification of the importance of some of these environmental conditions as seen from a software developer business perspective, including across the four dimensions cited in Figure 1.11. The complete list of responses is outlined in Table 1.5, whereas the most important factors are highlighted in Figure 1.12.

The results from the questionnaire, which is illustrative and not necessarily representative, suggest that firms may tend to view availability of trained human capital as a crucial factor for software development. Customer requirements are also highly ranked with a particular impact on customer demands regarding quality costs and security. Other factors that were often underscored as highly important are: IPR protection; application of technological standards; legal, regulatory and administrative environment; customers' requirements regarding interoperability and security issues. The remaining were considered of lesser importance, including customers financial strength and access to financing; technological infrastructure; language and culture; taxation and the physical distance to the customer.

Box 1.9. Software IPRs and innovation

Software-related innovations can be protected as intellectual property and, as is often the case with intellectual property in other fields of technology, remain vulnerable to imitation. To maintain incentives to innovate, governments have developed a variety of means to protect the rights of innovators in the software sector. International agreements, either bilateral or multilateral, provide a basic framework for the protection of IPRs in the software sector, including in some cases standards for minimum protection.

The specific mix of protections available varies among countries, but copyright protection is generally afforded to software innovators, as is some form of patent protection for software-implemented inventions. Other types of protection are also available to software innovators, such as protection for trade secrets or trademarks. A broad spectrum of software innovators has come to rely on IPR protection as an integral part of their business strategies; this includes most producers of both proprietary and open source software and firms or individual programmers operating under a wide range of business models.

Under current international arrangements, copyright protection for software innovations is quite broad and in theory available in countries representing the vast majority of the world economy. Members of the World Trade Organisation (WTO, with 151 member economies at the time of writing) are subject to the Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPS), which provides for computer programs to be treated as literary works, eligible for protection under the Berne Convention for the Protection of Literary and Artistic Works (administered by the World Intellectual Property Organisation, WIPO). Under these accords, copyrights automatically come into effect with the creation of the work and generally benefit from minimum standards of protection[53] and national treatment in other signatory countries.[54] Signatories commit to recognise and enforce this protection internationally.

Concerning patents, the WTO TRIPS Agreement states that patents shall be available in all fields of technology, a provision that applies in principle to software-implemented innovations. Other international agreements, including free trade agreements, are also relevant to the patentability of software. Yet, the detailed manner of implementation of patent protection for software-implemented innovations is not prescribed in full under these international agreements. Consequently, there is some variability in the processes for obtaining patent protection, and in the scope of patentable subject matter. In particular whereas in certain economies (*e.g.* US) software is often protected with patents, there are economies (*e.g.* European Union) where the existing legal regulations limit software patentability or exclude it altogether.

Some observers consider patents and copyrights as complementary tools for the protection of intellectual property with respect to software (Einhorn, 1990): copyrights protect original computer programmes against unauthorised copying, whereas patents can be used to protect inventions (especially the underlying technical ideas and principles). Key differences arise in the characteristics of copyrights and patents: copyrights protect the work internationally without formalities, whereas patent protection is only granted in countries where the rights holder has applied for a patent (in certain economies patents cannot be applied to software, *e.g.* European Patent Convention excludes patents on programs for computers as such). In practise, the duration of protection for a copyright is longer (generally amounting to 70 years or author's life plus 70 years), whereas patents generally expire 20 years from the date of application.

Table 1.5. Perceived importance of environmental factors to software developer operations

Percentages

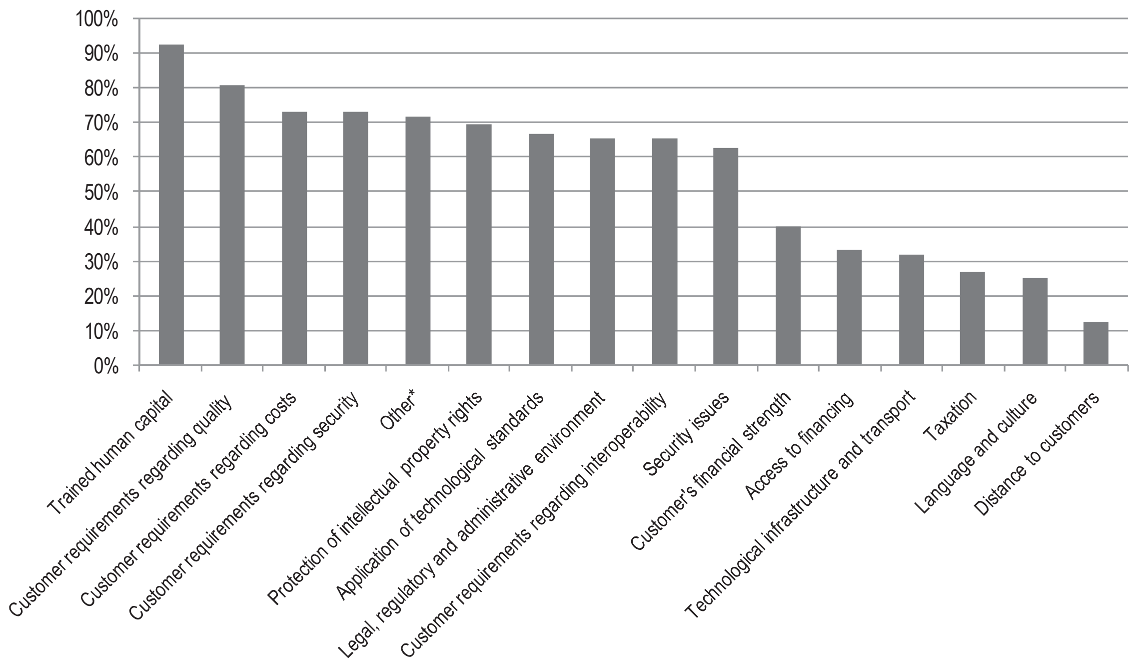
Factor	Importance				
	High	Medium	Low	Not important	Unknown
Trained human capital	92	8	0	0	0
Customer requirements regarding quality	81	15	4	0	0
Customer requirements regarding costs	73	19	8	0	0
Customer requirements regarding security	73	19	8	0	0
Other*	71	14	0	0	14
Protection of intellectual property rights	69	15	8	0	4
Application of technological standards	67	29	4	0	0
Legal, regulatory and administrative environment	65	15	19	0	0
Customer requirements regarding interoperability	65	31	4	0	0
Security issues	63	33	4	0	0
Customer's financial strength	40	44	8	0	4
Access to financing	33	42	21	4	8
Technological infrastructure and transport (*e.g.* broadband coverage)	32	44	20	4	0
Language and culture	25	46	21	8	0
Taxation	27	31	19	15	8
Distance to customers	13	50	38	0	0

Notes: N = 26. The figure presents the share of firms identifying each factor as being of the indicated degree of importance to the firm's software development operations.

Source: OECD (2008a), business questionnaire and related responses.

Figure 1.12. Factors of high importance for innovation, as assessed by software developers

Percentage of replies; respondents could indicate multiple factors



Notes: * Other factors identified by developers.
N = 26. The figure presents the share of firms identifying each factor as being of high importance to the firm's software development operations.

Source: OECD (2008a), business questionnaire and related responses.

Notes

1. The software field has a particular jargon and definitions for particular terms may vary, which can be confusing. A variety of easy-to-understand glossaries exist to help guide those new to the field. See, for example, Bixler and Bergman (1997) or the UNESCO *Learning without Frontiers* website: *www.unesco.org/education/educprog/lwf/doc/portfolio/definitions.htm*.

2. In some cases, alternatively, the source code can be executed directly using an intermediate computer program called an interpreter.

3. The term open source software (OSS) is often used loosely to refer to the three dimensions of openness. First, it points at the openness of the source code of software, as opposed to the case of closed source where the source code is not revealed publicly. Second, openness can refer to the openness of gratis distribution. Third, openness can refer to openness to improvement. In the context of this study the term OSS also encompasses so-called "free software" (a definition of the latter concept can be found on the GNU project website: *www.gnu.org/philosophy/free-sw.html*).

4. Not all electronic content can be classified as software. In particular, it is important to distinguish software from data. While a software program commands a physical device (hardware), data contain only given information and are not designed to control or instruct any hardware.

5. For example, Moore's Law holds that the density of transistors on an integrated circuit doubles every 18 months, growing as rapid geometric progression and providing more processing speed and possibly expanded functionality. Other dimensions of hardware are also experiencing rapid change and may reach a similar pace of evolution as a consequence of increased integration (IEEE, 2006).

6. For example, one traditional classification of the sector used by some public agencies for statistical purposes, focussed on just three types of software that could be readily measured (packaged, in-house and custom); but in the current environment, this is inadequate for describing the variety of new forms and modalities operating in the sector such as software as a service (described below).

7. PC Pro magazine offers research on an list of applications software, which is illustrative of the diversity of application software: barcode label, communications, consumer applications, contact management, diagnostic and testing, enterprise applications, geographical information, graphics, health care, industrial applications, insurance applications, interactive media applications, internet applications, legal applications, mathematical, network applications, office applications, portal applications, project management, publishing, scientific applications, simulation and analysis and wireless applications.
See: *http://research.pcpro.co.uk/rlist/term/Applications-Software.html* (last accessed 26 July 2008).

8. Examples of middleware initiatives include Open Software Foundation's Distributed Computing Environment (DCE), Object Management Group's Common Object Request Broker Architecture (CORBA), and Microsoft's COM/DCOM (Component Object Model (COM), DCOM, and Related Capabilities),
http://www.sei.cmu.edu/str/descriptions/middleware.html (last accessed 19 May 2008).

9. From information on the 1st International Conference on Middleware Security (2008), *http://www.cs.kuleuven.be/conference/MidSec2008/* (last accessed 26 May 2008).

10. Sometimes, firms or on-line communities make reference to closed, cloud-based systems. But, for purposes of this paper, the cloud is seen as a universal concept encompassing the entire Internet. Bounded cloud-based systems are seen as subsets of the whole.

11. A similar definition was proposed by McNee (2007) who defines SaaS in terms of the functionality of the application or utility infrastructure software that is provided by a SaaS supplier over the network.

12. A related concept is service oriented architecture (SOA), whereby enterprises can structure their computer systems in modules that can be accessed in various combinations and by various users as necessary to deliver services. IBM, for example, offers services to transform business legacy systems using SOA approaches as a way to update them and extract value.

13. For example, a March 2008 abstract from Forrester points to rapid growth in this area: *www.forrester.com/Research/Document/Excerpt/0,7211,44254,00.html*.

14. According to the SIIA (2001), the term ASP was coined by IDC and referenced in the paper "ASPs Are for Real … But What's Right for You?" an IDC White Paper (referenced at *http://whitepapers.zdnet.co.uk*).

15. PC Magazine, (2008), *www.pcmag.com/encyclopedia_term/0,2542,t=CRM&i=40485,00.asp*, (last accessed 4 July 2008).

16. This quote is from the description of embedded software provided by the Embedded Software Center of the University of Texas at Dallas, *http://esc.utdallas.edu/overview.html* (last accessed 1 August 2008).

17. The METI survey covered enterprises in the field of embedded software and included valid responses from 311 operational departments in 293 enterprises, 525 projects and 769 embedded software engineers.

18. This is despite progress in the System of National Accounts (1993 manual), which moved to better reflect software, stating: "Computer software that an enterprise expects to use in production for more than one year is treated as an intangible fixed asset. Such software may be purchased on the market or produced for own use. Acquisitions of such software are therefore treated as gross fixed capital formation. Software purchased on the market is valued at purchasers' prices, while software developed in-house is valued at its estimated basic price or at its costs of production if it is not possible to estimate the basic price. Gross fixed capital formation in software also includes the purchase or development of large databases that the enterprise expects to use in production over a period of time of more than one year. These databases are valued in the same way as software" (paragraphs 10.92 and 10.93).

19. According to OECD, gross fixed capital formation as defined by the European System of Accounts "consists of resident producers' acquisitions, less disposals, of fixed assets during a given period plus certain additions to the value of non-produced assets realised by the productive activity of producer or institutional units." OECD on-line *Glossary of Statistical Terms*, available at *http://stats.oecd.org/glossary/detail.asp?ID=1173* (accessed 2 August 2008).

20. According to the OECD's on-line *Glossary of Statistical Terms*, capital services "reflect a (physical) quantity, not to be confused with the value, or price concept of capital. Capital services are the appropriate measure of capital input in production analysis." The glossary is available here: *http://stats.oecd.org/glossary/detail.asp?ID=270* (accessed on 2 August 2008).

21. The sample included data for personal computer operating systems and productivity suites advertised in PC World magazine by retail vendors during the time period 1984 to 2000.

22. For an example of such data on applications employed in enterprises, see Saugatuck Technology (2007).

23. Survey-based results on market penetration are not free from certain risks and biases, particularly where they cannot be automated and depend on knowledgeable respondents. They may fail to take into account the intensity of use by respondents with respect to the various software products (*e.g.* in cases where there are competing products held by an individual user). In cases where surveys focus on distribution, the availability of free software may lead to wrong impressions in that the number of downloads may be much greater than the number of actual users. In addition, some of the surveys are based on relatively small samples of firms or users and face sample-selection problems. Another issue related to user surveys concerns the potential respondent bias. According to the results of the FLOSSPOLS survey, software users are very often unaware of the openness of software they use. In fact 29.9% of OSS users claimed not to use "open source software" but reported they use "Linux", "Apache" or other open source software products. Such lack of awareness could translate into biased survey results. (The FLOSSPOLS interviews were made in 2004 among employees of 955 public sector organisations in 13 EU countries. The results are available at *http://flosspols.org*.)

24. See *http://news.netcraft.com/archives/web_server_survey.html*.

25. During this period, the change in employment in the sector was mixed across the EU15 countries with some seeing a decline in the share (*e.g.* France, Ireland and Austria). A portion of these declines could potentially be related to the reported increase in the offshoring of ICT-related activities. For further OECD research on the potential offshoring of ICT-intensive occupations see van Welsum and Vickery (2005) and van Welsum and Reif (2006a, 2006b). Another explanation for diverging employment trends may relate to differences in the rate of adoption and integration of new technologies including the automation or digitisation of tasks.

26. For review of OECD countries human capital development policies, see:
www.oecd.org/education and
www.oecd.org/findDocument/0,3354,en_2649_39263238_1_1_1_1_37455,00.html.

27. In an article entitled "Reaching for the Stars: Who Pays for Talent in Innovative Industries" Andersson *et al.* (2008) present the following conclusion: "software firms that operate in software sectors with highly skewed returns to innovation, or high upside gains to innovation, are more likely to attract and pay for highly talented workers. Such firms do so first by paying more upfront in starting salaries to attract skilled employees and second by rewarding workers handsomely for experience or loyalty."

28. Recent studies (*e.g.* Maurer and Scotchmer, 2006; Bitzer *et al.*, 2007; Feller *et al.*, 2005; and MERIT, 2006) have identified a variety of such motives that provide incentives for developers of such products.

29. For a general introduction to this perspective, see Mas-Colell, Whinston, and Green (1995), Kreps (1990) and Varian (1992). For more specific modelling techniques, see Grossman and Helpman (1991a, b) and Aghion and Howitt (1992).

30. For example, in the United States, the US government is the principal funder of basic research, while industry funds substantial portions of applied research and product development.

31. For more information on R&D strategies, see Chapter 2.

32. Network effects occur if the number of other customers who use a given product, matter for the value of that particular product to a potential customer (Shy, 1996, 2001). In the software

sector network effects occur when use of a particular program one individual indirectly increases the value of this program for other users. For example, an additional user of a particular messenger programme (*e.g.* Google Talk or Skype) increases indirectly its value for all other potential users, for whom the number of current clients is a component that affects the product's value.

33. This section draws heavily on the *OECD Science, Technology and Industry Outlook 2008* (OECD, 2008b).

34. Examples of partnerships with universities include: Oracle and CERN (European Organisation for Nuclear Research) for grid computing technologies; Microsoft, Nokia, Hitachi, and Toshiba with research centres at the University of Cambridge; Fujitsu collaborating with the Universities of Tokyo and Cambridge on quantum technologies.

35. Costs for research conducted under such approaches can be comparatively low in cases where some of the participants are motivated by incentives other than direct monetary interest in a specific commercial product. This may take place for example when one of the parties has the expectation of eventually capitalising on the experience in developing a commercial product at a later stage (Iansiti and Richards, 2006).

36. The objective of the International Technology Roadmap for Semiconductors (ITRS) is to ensure advancements in the performance of integrated circuits and remove roadblocks to the continuation of Moore's Law. This assessment, called roadmapping, is a co-operative effort of global industry manufacturers and suppliers, government organisations, consortia, and universities. More information can be found on the ITRS website at *http://www.itrs.net/* (last accessed 27 September 2008).

37. Software as a product is used and developed not just for other industries, but also for the software industry itself. This overlap is often not taken into account in the statistics.

38. R&D strategies and the various research processes in the software sector are covered in detail in Chapter 2.

39. Also see the *IEEE Standard Glossary of Software Engineering Terminology*, IEEE std 610.12-1990, referenced at: *http://standards.ieee.org/reading/ieee/std_public/description/se/610.12-1990_desc.html*.

40. According to Donzelli *et al.* (2006), requirements engineering (concerned with methods, techniques and tools for eliciting, modelling and analysing software requirements) is one of the most critical areas of software development. They note that requirements errors are among the most costly and time-consuming to correct and that erroneous or omitted requirements are often indicated as the main reasons for project failures.

41. Examples of efforts to enhance productivity include the use of model-driven development, agile processes and SaaS.

42. Also, the development of parallel processing is contributing to expand the possibilities for software.

43. The Techweb Network offers the following explanation of the term *application framework*, "A set of common software routines that provides a foundation structure for developing an application. Frameworks take the tedium out of writing all the program code for an application from scratch. Object-oriented application frameworks, which are the norm today, are structured as a class library." The full explanation is available at: *www.techweb.com/encyclopedia/defineterm.jhtml?term=applicationframework* (last accessed 1 August 2008).

44. In some countries (*e.g.* Spain), governments have supported software development frameworks with a view to promoting openness.

45. Illustrative examples include:

1. Sun's *Java Native Interface, http://java.sun.com/j2se/1.3/docs/guide/jni/*

2. Microsoft's *.NET Framework, www.microsoft.com/downloads/details.aspx?familyid=333325FD-AE52-4E35-B531-508D977D32A6&displaylang=en*

3. Salesforce.com's *Application Framework*, which the firm claims will deliver "the ability to customise existing applications or build applications from scratch without writing any code." *http://wiki.apexdevnet.com/index.php/Application_Framework*

4. Apache claims that its *Struts* framework aims "to leverage existing standards by producing the missing pieces we need to create enterprise-grade applications that are easy to maintain over time. […] Struts 1 is recognised as the most popular web application framework for Java", *http://struts.apache.org/.*

Websites cited above last accessed 3 August 2008.

46. Parker and Van Alstyne (2008) provide a useful assessment of the economic context for platforms.

47. The discussion forum can be found here: *http://www.linkedin.com/answers/technology/enterprise-software/TCH_ENT/47038-3420784?browseIdx=0&sik=1183416866316&goback=.amq* (last accessed 7 May 2008).

48. Here it should be noted that some open source developers use intellectual property protection as a means of specifying the permitted uses of the open software. For example, intellectual property can also used to ensure openness in subsequent applications of the code.

49. For more on open business models see *http://rsss.anu.edu.au/~janeth/OSBusMod.html*, (last accessed 7 August 2008).

50. For example, Microsoft developed and actively uses two open source licenses approved by the Open Source Initiative – the Microsoft Public License and the Microsoft Reciprocal License – as well as a variety of other licensing schemes that allow users the right to redistribute or modify the original software.

51. Many business models can also be employed using closed source software or a hybrid approach.

52. For example, in the United States, the US government is the principal funder of basic research, while industry funds substantial portions of applied research and product development.

53. For example, the Berne Convention provides for a minimum term of protection.

54. National treatment in effect means that, for example, works that have a country of origin that is a Berne Union country benefit in all other Union countries from the same protection as the latter give to the works of their own nationals.

References

Aghion, P., and P. Howitt, (1992) "A Model of Growth through Creative Destruction", *Econometrica*, 60.

Ahmad, N. (2003), "Measuring Investments in Software", OECD Directorate for Science, Technology and Industry Working Paper 2003/6, OECD, Paris, available at *www.oecd.org/sti/working-papers*.

Andersson, F., J.C. Haltiwanger, M. Freedman, J. Lane and K.L. Shaw (2008), 'Reaching for the Stars: Who Pays for Talent in Innovative Industries?, via the Social Science Research Network, available at: *http://papers.ssrn.com/sol3/papers.cfm?abstract_id=918000*.

Bell, M. (2008). "Introduction to Service-Oriented Modeling" in *Service-Oriented Modeling: Service Analysis, Design, and Architecture*, Wiley & Sons.

Bernstein, P.A. (1996), "Middleware: A Model for Distributed Services." Communications of the ACM 39, 2, February.

Bitzer, J., W. Schrettl and P.J.H. Schroeder (2007), "Intrinsic Motivation in Open Source Software Development", *Journal of Comparative Economics*.

Bixler, B. and T. Bergman (1997), "Selecting and Implementing Computer-based Training", National Alliance of Business, Washington, DC.

BLS (2008), *Occupational Outlook Handbook, 2008-09 Edition*, Computer Software Engineers, Bureau of Labor Statistics, US Department of Labor, *www.bls.gov/oco/ocos267.htm*.

BSA (2007b), Global Piracy Study. Business Software Alliance, available at: *http://w3.bsa.org/globalstudy*.

Campbell-Kelly M. and D.D. Garcia-Swartz (2008), "Pragmatism not Ideology: IBM's Love Affair with Open Source Software", SSRN Working Paper, January 2008, *http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1081613*.

Carey (2008), Cloud Computing: The Evolution of Software-as-a-Service, Knowledge@W.P. Carey, Arizona State University, 4 June, *http://knowledge.wpcarey.asu.edu/article.cfm?articleid=1614*.

Chesbrough, H.W. (2003), *Open Innovation: The New Imperative for Creating and Profiting from Technology*, Harvard Business School Press, Boston.

Chesbrough, H.W., W. Vanhaverbeke and J. West (eds.) (2006), *Open Innovation: Researching a New Paradigm*, Oxford University Press, Oxford.

Donzelli, P., F. Shull, S. Asgari and V. Basili (2006), "Evolving a Dependability Requirements Elicitation and Modeling Framework Based on Use", Technical Report CS-TR-4851, University of Maryland, November, *www.lib.umd.edu/drum*.

Einhorn, D.A. (1990), “Copyright and Patent Protection for Computer Software: Are They Mutually Exclusive?”, *The Journal of Law and Technology*.

Enkel, E. and O. Gassmann (2004), “Driving Open Innovation in the Front End”, EURAM Conference Paper, 16 May, *www.alexandria.unisg.ch/Publikationen/37584*.

Fagerberg, J., D.C. Mowery and R.R. Nelson (2004), *The Oxford Handbook of Innovations*, Oxford University Press.

Feller, J., B. Fitzgerald, S.A. Hissam and K. Lakhani (2005), *Perspectives on Free and Open Source Software*, MIT Press, Cambridge, MA

Forrester (2006), Trends 2006: HR/HCM Applications Focus Shifts from Efficiency to Performance, Forrester Research, Inc., *www.forrester.com/Research/Document/Excerpt/0,7211,38498,00.html*.

Freeman, C. and L. Soete (2007), “Developing Science, Technology and Innovation Indicators: What We Can Learn from the Past”, Maastricht Economic and Social Research and Training Centre on Innovation and Technology, UNU-MERIT Working Paper 2007-001, *www.merit.unu.edu/publications/wppdf/2007/wp2007-001.pdf.*

FTC (2003), “To Promote Innovation: The Proper Balance of Competition and Patent Law and Policy”, report by the Federal Trade Commission, USG.

Gartner (2001), “Enterprise Applications: Adoption of E-Business and Document Technologies, 2000-2001: Worldwide Industry Study,” Report for AIIM International.

Gartner (2006), “Magic Quadrant for Integration Service Providers, 1Q06,” *www.e2open.com/news/article.php?item=214*.

Gartner (2007), Toolkit: How to Evaluate a Software Vendor’s Commitment to SaaS, ID Number: G00146039, Business Issues, Robert Desisto, 5 February.

Grimm, B., B. R. Moulton, and D. B. Wasshausen (2002), “Information Processing Equipment and Software in the National Accounts”, NBER Working Paper (prepared for conference on Measuring Capital in the New Economy), available at *www.nber.org*.

Grossman, G, and E. Helpman (1991a), “Quality Ladders in the Theory of Growth”, *Review of Economic Studies*, vol. 58(1).

Grossman, G, and E. Helpman (1991b), “Quality Ladders and Product Cycles”, *Quarterly Journal of Economics*, vol. 106 (425).

Harter, D.E., M.S. Krishnan and S.A. Slaughter (2000), “Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development”, *Management Science*, vol. 46, no. 4.

Hope J. (2003), “Open Source as a Business Approach”, Open Source Biotechnology Project, Research School of Social Sciences, Australian National University, Canberra, *http://rsss.anu.edu.au/~janeth/OSBusMod.html*.

Iansiti, M. and G.L. Richards (2006)," The Business of Free Software: Enterprise Incentives, Investment, and Motivation in the Open Source Community", Harvard Business School Working Paper No. 07-028.

IEEE (2004), Guide to the Software Engineering Body of Knowledge (SWEBOK), A. Abran and J.W. Moore (executive eds.) and P. Bourque and R. Dupuis (eds.), Los Alamitos, CA, USA. (SWEBOK is also published as ISO/IEC TR 19759:2005.)

IEEE (2006), IEEE Spectrum, June 2006", *www.spectrum.ieee.org/jun06/3649*.

ISO (2003), Proposed Draft Technical Report for Information Technology – Learning, Education and Training – Management and Delivery – Specification and Use of Extensions and Profiles, International Organisation for Standardization, http*://old.jtc1sc36.org/doc/36N0646.pdf.*

Jaumotte, F. and N. Pain (2005a), "An Overview of Public Policies to Support Innovation", OECD Economics Department Working Paper No. 456, OECD, Paris.

Jaumotte, F. and N. Pain (2005b), "From Ideas to Development: The Determinants of R&D and Patenting", OECD Economics Department Working Paper No. 457, OECD, Paris.

Jaumotte, F. and N. Pain (2005c), "From Development to Implementation: Evidence on Innovation Determinants from the Community Innovation Survey", OECD Economics Department Working Paper No 458, OECD, Paris.

Jaumotte, F. and N. Pain (2005d), "Innovation Policies and Innovation in the Business Sector", OECD Economics Department Working Paper No. 459, OECD, Paris.

Johnson, R. E. (1997) "Frameworks = (Components + Patterns)", Communications of the ACM, Vol. 40, Issue 10, *http://portal.acm.org/citation.cfm?id=262793.262799&dl=GUIDE&dl=ACM&idx=262793&part=periodical&WantType=periodical&title=Communications%20of%20the%20ACM.*

Johnston, S. (2008), "The Cloud and Cloud Computing: Consensus Definition?, blog entry, *http://samj.net/2008/07/cloud-and-cloud-computing-consensus.html*.

Knorr, E., and G. Gruman (2008), "What Cloud Computing Really Means", *The New York Times*, 7 April, *www.nytimes.com*.

Kreps, D.A. (1990), *Course in Microeconomic Theory*, Princeton University Press, Princeton, NJ.

Lassila, A. (2007) "Mobile Security as a Service", 40th Annual Hawaii International Conference on System Sciences, HICSS 2007, available at: *http://project.hkkk.fi/helsinkimobility/papers/Mobile%20Services_2_3.pdf.*

Lequiller, F., N. Ahmad, S. Varjonen, W. Cave and K. Ahn (2003), "Report of the OECD Task Force on Software Measurement in the National Accounts," OECD Statistics Working Papers, No. 2003/1, OECD Statistics Directorate.

Mas-Colell, A., M. Whinston, and J. Green. (1995), *Microeconomic Theory*, Oxford University Press, New York, NY.

Maurer, S.M. and S. Scotchmer (2006), "Open Source Software: The New Intellectual Property Paradigm", NBER Working Paper No. 12148.

McNee, W. (2007), "SaaS 2.0", *Journal of Digital Asset Management*, Vol. 3, *www.palgrave-journals.com/dam/journal/v3/n4/abs/3650088a.html*.

MERIT (2006), Economic Impact of FLOSS on Innovation and Competitiveness of the EU ICT Sector, UNU-MERIT, Maastricht University.

METI (2007), "Outline of Survey on the Situation in Embedded Software Sector for 2007" (in Japanese), news release, http://sec.ipa.go.jp/download/200706es.php.

NATO (1969), Software Engineering: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 October 1968, P. Naur and B. Randell (eds.), Brussels, Scientific Affairs Division, NATO, as reformatted in 2001, available as of 13 July 2008 at: *http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF*.

Netcraft (2008), Web Server Survey, available at*: http://news.netcraft.com.*

OECD (2001), "Inquiry Into National Practices in the Measurement of Software in the National Accounts and International Trade, Questionnaire – Software", OECD, Paris.

OECD (2002), *OECD Information Technology Outlook 2002*, OECD, Paris.

OECD (2004), *OECD Information Technology Outlook 2004*, OECD, Paris.

OECD (2005), O*slo Manual: Proposed Guidelines for Collecting and Interpreting Technological Innovation Data, 2nd edition*, OECD, Paris.

OECD (2006), *OECD Information Technology Outlook 2006*, OECD, Paris, *www.oecd.org/sti/ito*.

OECD (2007), OECD Productivity Database, OECD, Paris.

OECD (2008a), "Innovation in the Software Sector: Business Questionnaire" and associated responses.

OECD (2008b), *OECD Science, Technology and Industry Outlook 2008*, OECD, Paris, *www.oecd.org/sti/outlook*.

OECD (2008c), Workshop on New Forms of Entrepreneurship and Innovation, May, *www.oecd.org/sti/innovation/entrepreneurship*.

OECD (2009), OECD Productivity Database, OECD, Paris.

Parker, R. and B. Grimm (2000), "Recognition of Business and Government Expenditures for Software as Investment: Methodology and Quantitative Impacts, 1959-98", US Bureau of Economic Analysis Working Paper, available at *www.bea.gov*.

Parker, G. and M.W. Van Alstyne (2008), "Innovation, Openness, and Platform Control", 24 July, *http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1079712*.

Pree, W. and H. Sikora (1997), "Design Patterns for Object-Oriented Software Development, Proceedings of the 19th International Conference on Software Engineering",: *http://portal.acm.org/citation.cfm?id=253228.253810.*

Rajala, R., M. Rossi and V. Tuunainen (2003), "A Framework for Analyzing Software Business Models", *http://is2.lse.ac.uk/asp/aspecis/20030126.pdf*.

Saugatuck Technology (2007), "Open Source: The Next Disruptive IT Influence", *www.saugatech.com/395order.htm*.

Schreiber, R. (1995), "Middleware Demystified." *Datamation*, Vol. 41, No. 6, 1 April.

Shy, O. (1996), *Industrial Organization: Theory and Applications,* MIT Press.

Shy, O. (2001), *The Economics of Network Industries*, Cambridge University Press.

SIIA (2001), Software as a Service: Strategic Backgrounder, Software and Information Industry Association, February, *www.siia.net/estore/ssb-01.pdf*.

Silberschatz, A., P.B. Galvin and G. Gagne (2005), "Operating System Concepts, 7th Edition", slides available at *http://codex.cs.yale.edu/avi/os-book/os7/slide-dir/index.html*.

Software Magazine (2008), "The Software Top 500 Survey", available at *www.softwaremag.com.*

Taylor, R.N. and A. Van der Hoek (2007), "Software Design and Architecture: The Once and Future of Software Engineering", *Future of Software Engineering 2007*, IEEE.

UNCTAD (2002), "Changing Dynamics of Global Computer Software and Services Industry: Implications for Developing Countries", Technology Development Series, UNCTAD/ITE/TEB/12, United Nations Conference on Trade and Development, available *www.unctad.org/en/docs/psitetebd12.en.pdf*.

Van Genuchten, M. (2007), "The Impact of Softwar e on the Electronics Industry", *Computer*, IEEE.

Van Genuchten, M. (2008), "The Impact of Software Growth on Companies, Industries and National Economies", issues paper prepared for the meeting of the OECD Advisory Expert Group on Innovation in the Software Sector, Tokyo, 7 October.

Van Welsum, D. and G. Vickery (2005), "Potential Offshoring of ICT-intensive Occupations", in OECD (2005), *Enhancing the Performance of the Services Sector*, OECD, Paris, pp. 179-204, also available at *www.oecd.org/sti/offshoring.*

Van Welsum, D., and X. Reif (2006a), "The Share of Employment Potentially Affected by Offshoring: An Empirical Investigation", DSTI/ICCP/IE(2005)8/FINAL, Directorate for Science, Technology and Industry, OECD, Paris; available at *www.oecd.org/sti/offshoring.*

Van Welsum, D., and X. Reif (2006b), "Potential impacts of international sourcing on different occupations", DSTI/ICCP/IE(2006)1/FINAL, Directorate for Science, Technology and Industry OECD, Paris; available at *www.oecd.org/sti/offshoring*.

Varian, H. (1992), *Microeconomic Analysis*, W. W. Norton, 3rd ed. New York, NY.

White, A.G., J.R. Abel, E.R. Berndt and C.W. Monroe (2004), "Hedonic Price Indexes for Personal Computer Operating Systems and Productivity Suites", NBER Working Paper No. 10427.

Wintergreen, (2006), Wintergreen Research Group Report # SH29821464, *www.wintergreenresearch.com/reports/soa.html*.

Workday (2008), "From Applications to Services, the Shift to On-demand", White Paper, available at *www.cfo.com/whitepapers/index.cfm/displaywhitepaper/11778019*.

Young, A. (1996), "Measuring R&D in Services", STI Working Paper, OCDE/GD(96)132, Directorate for Science, Technology and Industry, OECD, Paris.

Chapter 2

R&D and the Market Environment in the Software Sector

This chapter explores the way R&D is handled by companies in the software sector in response to the rapidly changing market environment. The analysis is conducted in the light of the broader properties characterising the sector: how companies manage R&D, how they are funded, how they collaborate, how they use external resources, and so on. Additionally, this chapter provides four illustrative country case studies.

Introduction and main findings

The software sector is one of the sectors most dependent upon research and development (R&D) to fuel innovation. Although R&D and innovation[1] are closely related, they can and do diverge. Not all R&D will lead to innovation, and not all innovation requires R&D. Nevertheless, in general, the introduction of improved software functionality or quality requires greater expenditure on R&D and production (*e.g.* in terms of deeper testing in order to detect and fix potential problems). This is a costly endeavour, a point confirmed in the literature on the topic.[2] Since the marginal costs of software production are comparatively small, the main costs that software developers face in attaining appropriate quality standards are experienced in the R&D phase.[3]

This chapter looks at R&D and the market environment in the software sector. It examines R&D indicators, globalisation of software markets and R&D, R&D in a collaborative context and national frameworks for R&D. Building on a review of available evidence, this chapter highlights the importance of R&D processes in the software sector as a key contributor to the very dynamic pace of innovation in the sector, while also signalling evolution and change in those very processes. The analysis points to five main findings:

- *Increasing intensity and extensiveness of R&D*. The intensity of R&D activities in the software sector is high and tending to rise (*e.g.* in terms of key expenditure indicators). At the same time, the extent of R&D processes is broadening (*e.g.* geographically and in terms of the range of stakeholders directly involved). The combination is fuelling a significant increase in software R&D activities through internationalisation, collaboration and openness, among other developments. Also contributing to the increase in intensity and extensiveness of R&D activity is the market's focus on mixed-source or hybrid computing environments. Companies and governments no longer look to one vendor, or one business, licensing or development model, for all of their computing needs. Instead, they now look to multiple vendors for their hardware, software, and services. R&D into market-driven interoperability is a fast growing category.

- *Human capital is at the heart of the R&D process*. While human capital is generally important in the software sector (see also Chapter 1), it appears particularly important in the R&D activities of the sector. This involves the knowledge, skills and aptitudes of participants, and the way they interact. Networks are critical, both inside of multinational enterprises and among more disparate collaborators.

- *R&D processes take place in the context of dynamic interplay between technological progress and market demand*. As the software industry matures and technological progress continues (expanding the possibilities for new software functionality), the R&D processes increasingly take into account user input as well as the technical possibilities afforded by improved hardware capabilities. This dual approach may help to better ensure market-relevance of the output of R&D processes. For example, new technical possibilities via the Internet are being considered in light of user demand for secure, reliable, powerful and integrated systems, leading to new developments in software-as-a-service and, more broadly, cloud computing.

- *Tremendous potential for economies of scale via software R&D*. Globalisation, the mainstreaming of software throughout the devices associated with modern life, and the low marginal cost of reproduction of software mean that there are tremendous opportunities to leverage R&D. This can be via collaboration on the input side to reduce costs or expand the potential for results (*e.g.* better resulting functionality) and via reuse or broad application of the results on the output side (*e.g.* in many products across global markets). Ecosystems of participants are engaging in multidimensional co-operation to drive forward the R&D processes in the sector.

- *Much of this activity is taking place under the auspices of the private sector* (enterprises and other institutions). Still, from case studies and other evidence, it appears that policy can influence the environment for innovation in important ways, such as promotion of human capital development or basic research, or seeding development of quasi-public goods like the Internet.

R&D investment in the software sector

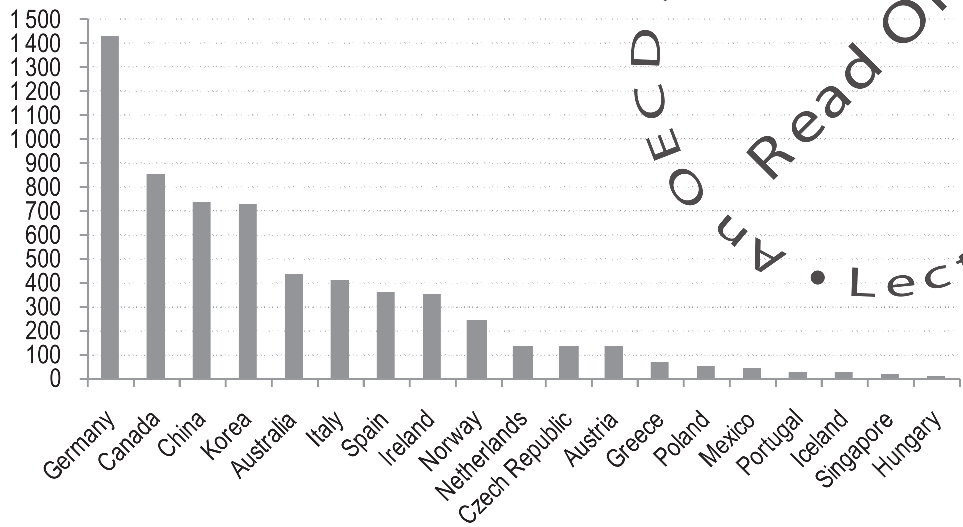
Existing data at the national-level and firm-level point to the large scale and intensity of software-related R&D activity, as well as its international scope. The following two sections treat these two levels, each in turn, followed by a discussion of software-related patents as one indicator of R&D output.

National-level estimates

The OECD Research and Development database contains data on Business Expenditure R&D (BERD),[4,5] which highlight the large scale of investment in the broad sector "computer and related activities" (Figure 2.1). The United States accounts – by far – for the largest absolute amount of expenditure on R&D for this area,[6] followed by Israel, Japan and Germany. The data for the subcategory "software consultancy and supply" are shown in Figure 2.2. Here again the United States is the leading country, followed by Germany and Canada. China, notably, is also highly placed terms of business expenditure in R&D, ranking fifth in computer and related activities and fourth in the latter software consultancy and supply.

Figure 2.1. Business expenditure on R&D, computer and related services[1,2]

USD millions PPP, current prices



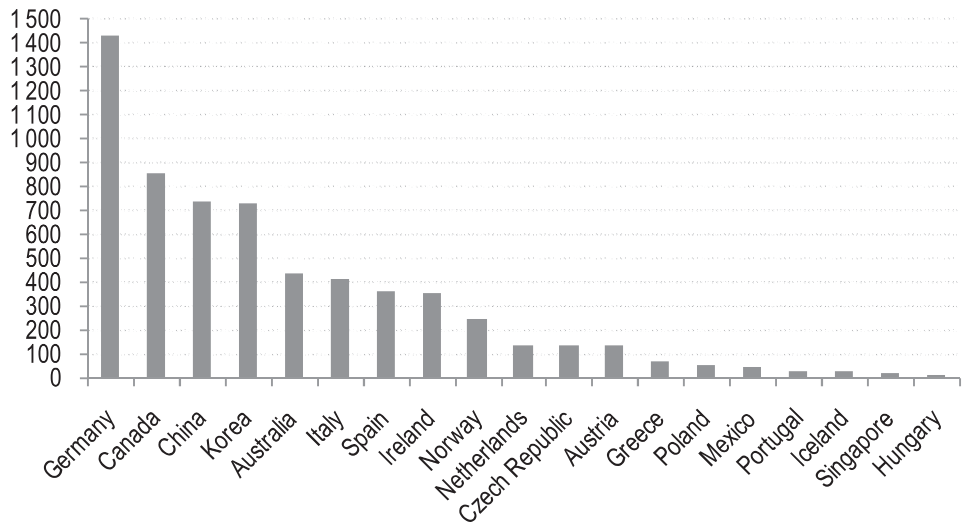
1. Data are for most recent year available, generally 2005, except as follows: Canada, Israel, Italy: 2006; Australia, Austria, Spain: 2004; Germany, Greece, Portugal: 2003; China: 2000.

2. The comparable figure for the United States is 22 265 for the year 2001.

Source: OECD Research and Development Database, preliminary.

Figure 2.2. Business expenditure on R&D, software consultancy and supply[1,2]

USD millions PPP, current prices



1. Data are for most recent year available, generally 2005, except as follows: Canada, Italy: 2006; Australia, Austria, Spain: 2004; Germany, Greece, Portugal: 2003; China: 2000.

2. The comparable figure for the United States is 13 111 for the year 2001.

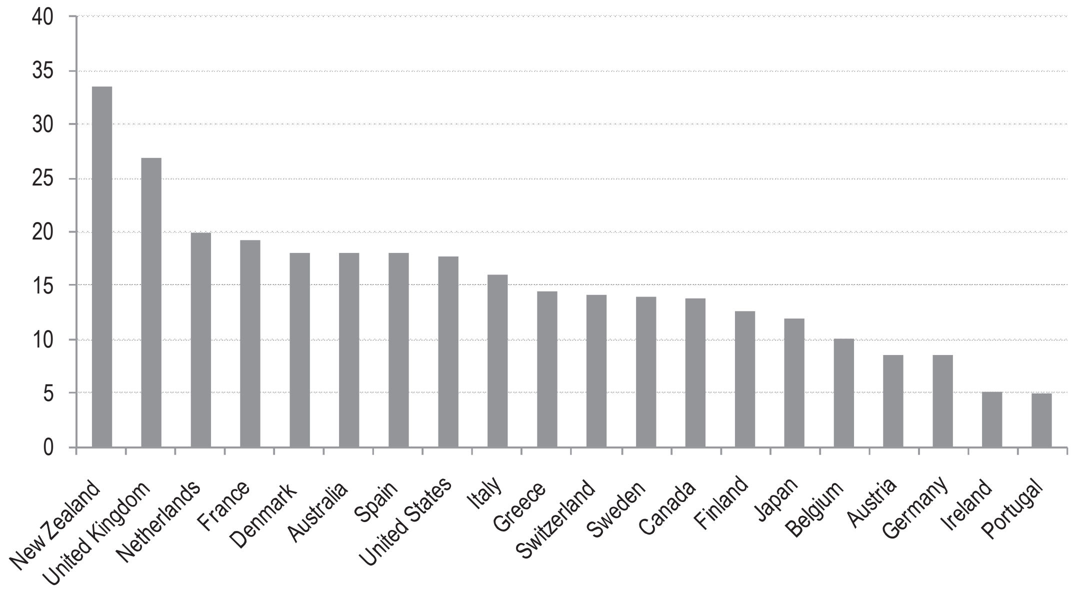
3. Data for Israel and Japan are not available for this subcategory. For Mexico, all of computer and related activities is comprised of software consultancy and supply. The percentage of computer and related activities accounted for by software consultancy and supply in the countries shown in Figure 2.2 varies between 100% (Mexico) and 19% (Singapore).

Source: OECD Research and Development Database, preliminary.

Another approach to estimate aggregate spending on R&D in software builds on a research finding from the literature whereby the share of software R&D in overall software investment is approximately 25% (Khan, 2004). Based on this estimate, the importance of R&D in software in total R&D is estimated to range from around 34% in New Zealand to around 5% in Ireland and Portugal (Figure 2.3). These estimates are illustrative of differences between countries in terms of the importance of software R&D in relation to their overall R&D efforts. That is, they highlight the relative importance of software R&D within a country's overall R&D investment, but do not consider the overall size of the investment in absolute terms.

Figure 2.3. R&D in software[1] as a share of total gross domestic expenditure on R&D[2,3]

(percentages)



1. R&D in software was estimated by the Secretariat as 25% of total software investment (based on a finding by Khan, 2004).

2. Gross domestic expenditure on R&D amounts to between 0.6% (Greece) and 3.5% (Finland) of GDP.

3. Data refer to 2005, except 2004 for Australia, Belgium, Greece, Japan, Switzerland and 2006 for Canada, France, Germany, Ireland, Sweden and the United States.

Source: OECD Capital Services Database.

Firm-level estimates

Although the data on R&D activity in the software sector are not always consistent in their coverage or treatment of software, they nonetheless paint a picture of a sector that his highly R&D focused.[7] Table 2.1 presents indicators based on firm-level data from a global R&D investment database sponsored by the European Commission (EC, 2007b,c).[8] As shown in Table 2.1, the broad category "software and computer services" places fifth among the 22 sectors represented by the top 1 400 companies in terms of absolute volume of R&D expenditure. In terms of the R&D expenditure as a percentage of sales (R&D intensity), the sector ranks second beyond pharmaceuticals and biotechnology. The R&D intensity of the sector is nearly three times the average for the top 1 400 firms as a group.

Table 2.1. R&D performance, by sector, top 1 400 companies, 2006

Top five sectors	Aggregate R&D investment (EUR millions)	Share in total R&D investment	R&D intensity (% of sales)
Pharmaceuticals and biotechnology	70 523.5	19.3	15.9
Technology hardware and equipment	64 531.5	17.6	8.6
Automobiles and parts	60 807.1	16.6	4.1
Electronic and electrical equipment	27 138.9	7.4	4.4
Software and computer services	26 522.8	7.3	9.8
Combined results, 22 sectors	**365 823.9**	**100.0**	**3.4**

Notes: The table is based on the top 1 400 companies in terms of absolute volume of investment in R&D during 2006. The categories are based on the ICB industry classification system. The category "software and computer services" refers to ICB 9530, which covers computer services, Internet services and software.

Source: EC (2007), *2007 EU Industrial R&D Investment Scoreboard*, European Commission, Joint Research Centre, Directorate General Research, October, Table 2.

Table 2.2 considers the top 150 software publishers in the world among the firms in the EC-sponsored R&D investment database (EC, 2007b). The table presents the average profitability of firms (as a percentage of sales) in relation to indicators of their R&D activity. The largest investors in R&D are the most profitable as measured by both aggregate R&D investment and R&D investment per employee. This does not imply causality (R&D could fuel innovation that drives profits or *vice versa*, or other factors may be at work). It does provide an indication that scale effects may play a role; large firms able to mobilise more resources for R&D may also turn out to be more profitable.[9]

Table 2.2. Profitability and R&D indicators for top software firms, ranked by volume of R&D investment, 2006

Firms	Average profitability (% of sales)	Average R&D intensity (% of sales)	Average R&D per employee (EUR thousands)	Average R&D expenditure (EUR millions)
Top 10	19.5	20.5	50.2	1132.4
Top 25	13.2	19.2	41.3	551.1
Top 150	3.7	20.1	31.5	118.2

Note: The data refer to ICB Sector 9537, software publishers and distributors of computer software for home or corporate use. They exclude Internet service companies and computer service companies as well as computer game producers.

Source: EC (2007), *2007 EU Industrial R&D Investment Scoreboard*, European Commission, Joint Research Centre, Directorate General Research, October, Table 2; authors' calculations.

Table 2.3 presents data on the leading firms (in terms of R&D investment) in software publishing, computer services and the Internet services (EC, 2007a). The firm-level data highlight the massive investments in R&D made by these leading firms. Moreover, many of the firms – particularly the Internet services and software publishing firms – have moved to substantially increase their R&D investment, with double-digit compound annual rates of growth (triple-digit growth in the case of Google).[10]

While the table reflects the strong presence of American firms among these R&D leaders, it is important to note that these data do not give an indication of the geographic location of the R&D investments; the investments of all establishments are attributed to the firm and the country of registration of its headquarters irrespective of where the actual investment took place. Nevertheless, the data seem to suggest some role for the domiciliation of the firms. In 2006 (EC, 2007b), only seven of the top 25 software publishing firms in terms of R&D investment were headquartered outside of the United States: one in Germany, three in France and three in the United Kingdom. Similarly, among the top 25 firms in the database in terms of net sales in 2006, only eight were headquartered outside of the United States.

Table 2.3. R&D investment by top companies operating in the fields of software, computer services and Internet, 2005

Company	Country	R&D investment		Net sales		Employee		R&D intensity	
		EUR (millions)	CAGR 3 years (%)	EUR (millions)	CAGR 3 years (%)	Number	CAGR 3 years (%)	R&D as % of sales	R&D per employee (EUR thousands)
Software									
1. Microsoft	USA	5 581.52	12.2	37 540	11.2	71 533	9.2	14.9	78.0
2. Oracle	USA	1 586.97	16.6	12 191	14.9	56 133	11.4	13.0	28.3
3. SAP	DEU	1 088.63	6.2	8 512	4.7	34 550	5.3	12.8	31.5
4. CA	USA	662.09	3.5	3 226	6.9	16 000	0.0	20.5	41.4
5. Symantec	USA	578.27	51.2	3 513	43.3	16 000	55.0	16.5	36.1
6. Cadence Design Systems	USA	358.88	3.0	1 127	0.9	5 000	-1.1	31.8	71.8
7. Adobe Systems	USA	309.70	14.1	1 667	19.1	5 734	19.7	18.6	54.0
8. Sega Sammy	JPN	298.73	n.a.	3 704	n.a.	5 407	n.a.	8.1	55.2
9. Intuit	USA	285.34	14.4	1 727	14.4	7 000	2.5	16.5	40.8
10. Synopsys	USA	271.27	12.1	841	3.0	4 756	3.8	32.3	57.0
Computer services									
1. IBM	USA	4 559.15	4.3	77 258	3.9	329 373	1.4	5.9	13.8
2. Unisys	USA	330.28	-1.9	4 882	0.9	36 100	-0.3	6.8	9.1
3. SunGard Data Systems	USA	207.53	12.0	3 318	15.7	15 000	19.5	6.3	13.8
4. DST Systems	USA	110.88	-3.7	2 132	1.8	10 500	-3.5	5.2	10.6
5. Indra Sistemas	ESP	85.90	6.0	1 202	11.2	7 584	7.6	7.1	11.3
Internet									
1. Google	USA	508.23	145.5	5 204	140.7	5 680	n.a.	9.8	89.5
2. Yahoo!	USA	498.07	56.0	4 457	76.7	9 800	39.6	11.2	50.8
3. Check Point Software Technologies	ISR	42.85	20.7	491	10.7	1 414	5.5	8.7	30.3
4. United Online	USA	33.92	17.3	445	46.3	900	28.9	7.6	37.7
5. F5 Networks	USA	26.58	20.3	239	37.5	792	n.a.	11.1	33.6

Notes: n.a. = not available; CAGR = compound annual growth rate. The table refers to the top companies in terms of expenditure on R&D.

Source: EC (2007), *2006 EU Industrial R&D Investment Scoreboard*, European Commission, Joint Research Centre, Directorate General Research, Technical Report EUR 22348.

Indicators for ICT and R&D[11]

The classification "ICT" includes software, but also a number of neighbouring sectors. That is, ICT is a broader classification. Nevertheless, in the absence of internationally comparable data on some dimensions of R&D, consideration of indicators for the broader sector can at least provide an indication of the environment in which the software sector operates. For example, it can help to shed some light on the case of employment in R&D activities in the "neighbourhood" of the software sector and on the case of publicly funded research.

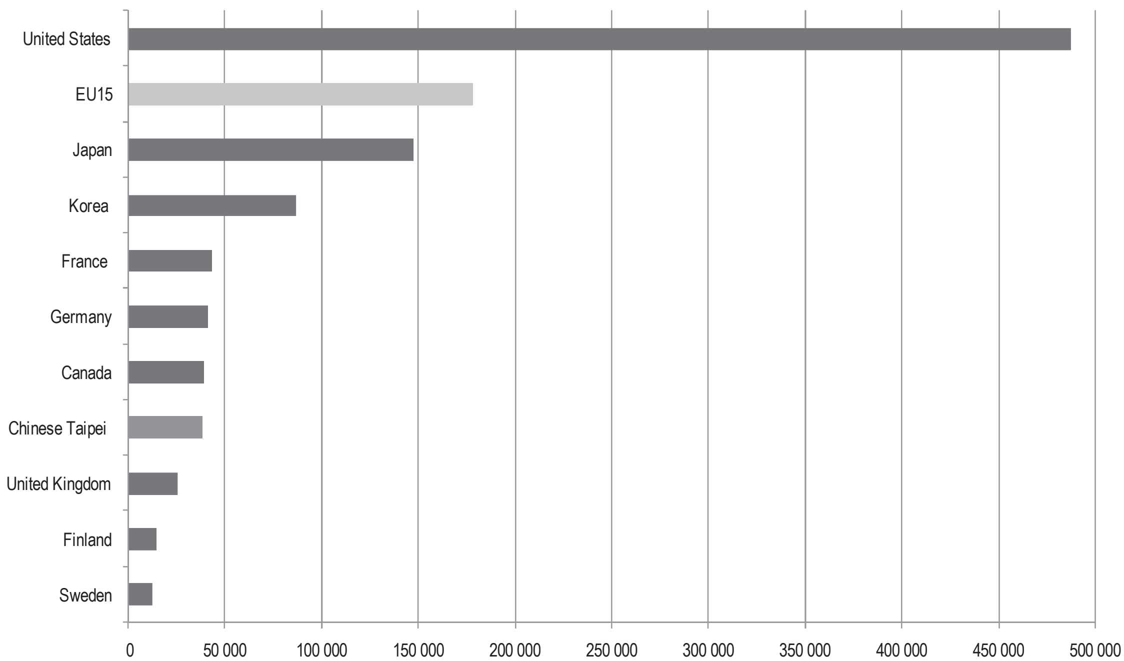
ICT employment in R&D activities

As noted in Chapter 1, the software sector in general depends heavily on its human capital. There are indications that this is particularly true with respect to R&D in the sector, despite the scarcity of internationally comparable data. One indication can be found in data for the broader ICT sector, of which the software sector is a part.[12]

R&D employment data for the ICT sector reflect the large scale of human resources deployed in these activities; across 25 developed countries in 2006, ICT R&D employment amounted to 943 000.[13] With nearly 487 000 researchers, the United States accounted for just over one-half of the total (Figure 2.4), followed by EU17 countries, Japan and Korea.[14] Figure 2.5 highlights the share of ICT R&D workers in the total for each economy. The OECD countries with the largest share of ICT R&D personnel by this measure are Ireland (54%), Korea (53%), Finland (51%), Denmark (39%), and Canada (39%). Among non-OECD economies Chinese Taipei (68%) and Singapore (40%) exhibit a relatively high degree of specialisation of their researchers in ICT.

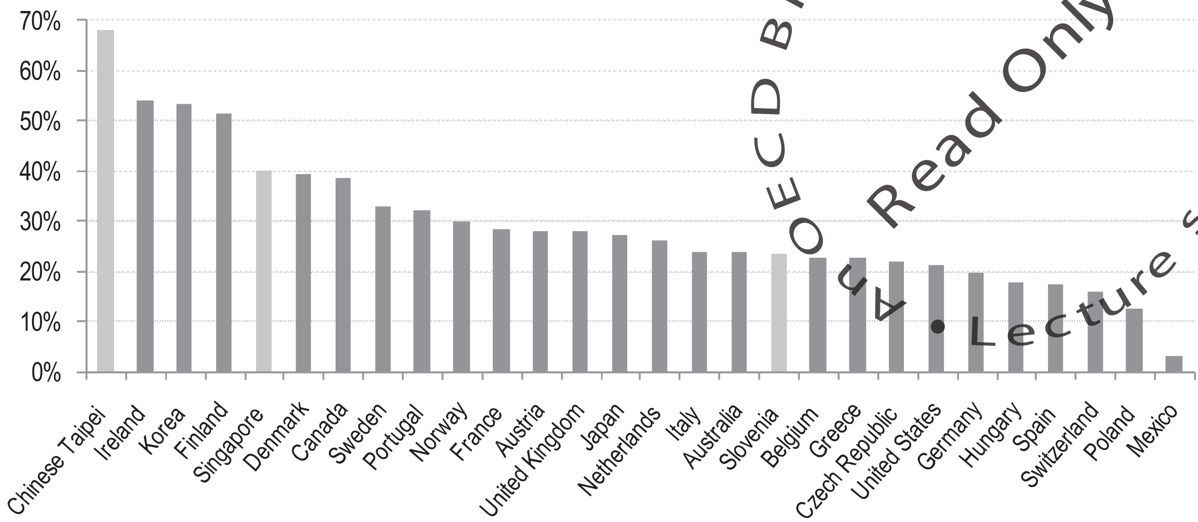
Figure 2.4. ICT R&D researchers 2006 or latest available year

Full-time equivalents



Source: OECD (2008), "Intellectual Assets and Value Creation: Synthesis Report", OECD, Paris, available at *www.oecd.org/sti/ipr/iavc.*

Figure 2.5. Share of ICT R&D researchers in total R&D researchers, 2006 or latest available year



Source: OECD (2008), "Intellectual Assets and Value Creation: Synthesis Report", OECD, Paris, available at *www.oecd.org/sti/ipr/iavc*.

ICT and the continuing importance of publicly funded research

When it comes to ICT inventions, the move from initial basic research to applied use can take decades. In some cases, unanticipated research results from fundamental research can provide important building blocks for subsequent innovation. Publicly-funded research has long been a stimulus for related business R&D and has contributed to the development of key technologies such as semiconductors or networking technologies (NRC, 2003; MIC Japan, 2005).[15] In particular, national space and defence R&D programs have funded a significant amount of ICT research, including in relation to hardware and software. Other examples of innovations drawing on publicly-funded basic research include the Internet, graphical user interfaces, global positioning systems and web search technologies.

The ICT sector has relied to some extent on publicly-funded R&D and on complex partnerships among government, public sector research organisations and industry, as well as long-term fundamental scientific research performed at universities.[16] Firms conducting ICT-related research often cluster close to universities, benefitting from the positive spill-overs associated with public ICT-related R&D. In the United States, for instance, 70% of R&D performed by all domestic and foreign computer and electronic firms in 2005 was located in four locations[17] with proximity to public research institutes and major universities (National Science Board, 2008).

In recent years, OECD member countries have substantially increased overall public funding for R&D (OECD, 2008c). Data on government budget appropriations or outlays for R&D show that between 2000 and 2006, government R&D budgets in the OECD area expanded by 6.8% annually; this rate is greater than the rate of GDP growth. However, there is considerable inter-country variation in terms of the level and composition of R&D funding.[18] For example, the defence R&D budget of 0.6% of GDP in the United States is well in excess of those in United Kingdom or France (both at 0.2% of GDP).

Despite the importance of public research to the ICT sector, official figures on total funds allocated to publicly-funded ICT-related R&D are not available in an internationally comparable manner.[19] Public R&D spending data (*i.e.* government appropriations for ICT-related R&D) generally do not permit assessments specific to the ICT sector or detailed socio-economic objectives specific to ICTs. Although broad socio-economic objectives such as "Non-oriented research in Mathematics and Computer Science" are directly or indirectly related to ICT research, exact ICT-related figures are hard to produce.

While comprehensive data are not available, OECD governments are formulating ICT R&D funding programmes to promote research and (international) co-operation between the private and public sector. ICT R&D budgets in the United States (NITRD), Japan (Council for Science and Technology Policy's ICT-related R&D budget), and at the EU-level (ICT-related funding in FP7) are particularly notable in terms of absolute spending (over USD 1 billion annually).[20] In certain other countries, funding programmes for ICT research also receive high priority, either as stand-alone policies (*e.g.* Germany's ICT 2020) or as a major pillar of wider science, technology and innovation policies (*e.g.* Spain's "Ingenio 2010" and Canada's "Mobilizing Science and Technology to Canada's Advantage"). Most of these programmes represent only a subpart of total public funding available to ICT-related research and are often complemented by public research funding on a sub-national or national level. General public policy measures (*e.g.* R&D tax concessions in various forms) are also important for the promotion of ICT-related research. Non-OECD countries such as China and India are also increasing support for ICT-related research. These programmes are usually part of national science and technology agendas, but in comparison to the United States, Japan and the EU, absolute annual funding for explicit ICT-related research in non-OECD countries is still relatively low.

Finally, national ICT diffusion strategies entail plans to boost ICT R&D as a driver of innovation. Examples include the u-Korea Master Plan (2006-2010) and the Information Society Strategies in Turkey and Switzerland. Non-OECD member economies are also embedding R&D promotion into national ICT strategies (*e.g.* Singapore's Intelligent Nation 2015).

R&D and software-related patents[21]

Patent data can be considered as providing an indication of the output from R&D activity. However, the links between patenting and commercial innovation are complex and subject to variation across time, business environments and subject areas, among other dimensions. There is not necessarily a one-to-one relationship. Thus, patents are an imperfect indicator of innovative activity in the software sector.

A variety of issues may arise in the use of patents as such an indicator. One concerns the pace of innovation in the sector. Patent applications take substantial amounts of time for review and approval by the authorities, which may prompt some software innovators to rely on other more-expeditious means of protecting their innovative advantage (*e.g.* copyrights, trade secrets or time-to-market advantages in their innovative processes). The utility of these alternative approaches may vary depending on the market environment and nature of the innovation. Hence, software innovators may not behave consistently in their recourse to patent protection. Furthermore, the increased importance of open innovation and open source software may influence the use of patenting. Innovators using open approaches may invoke intellectual property rights and employ various licensing schemes (*e.g.* to defend the open nature of their creations), but they appear to have a much lower propensity to seek software-related patents than innovators using, for example, proprietary or hybrid approaches. There is also an issue of patent content; the number of patents granted does not provide an estimation of the quality or impact of the associated innovations. Finally, international comparisons are complicated by the fact that the treatment of software-related inventions may differ in the details across various countries; the ease with which firms may obtain patents for software-related innovations also varies among countries.

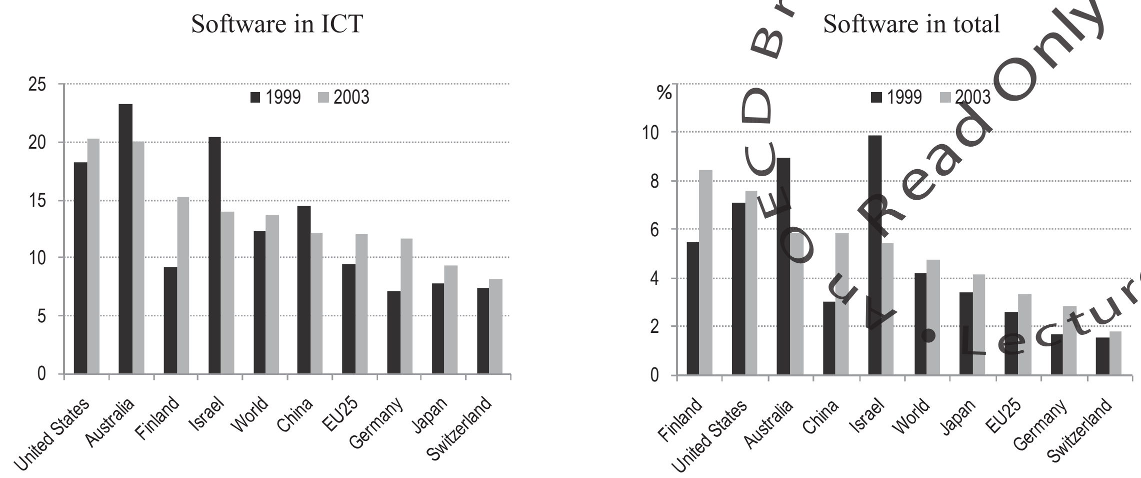
Despite the measurement challenges, research has shown that the stock of patents is correlated with firm success in the software industry (Merges, 2006). For example, use of patents may boost innovation in software by increasing possibilities to appropriate returns from R&D, raising productivity of R&D, and yielding more efficient transactions in knowledge (*e.g.* Smith and Mann, 2004). Software-related patents have been found to be useful for new, small, entrants to the software sector. Such firms may use patents to signal their expertise to third parties, negotiate cross-licensing arrangements, increase their value to potential buyers and convert tacit knowledge into a verifiable and transferable form (Mann, 2004).[22] Patents may help small firms attract venture capital or other finance, acting as a signal to potential investors. Patents can also facilitate intra-industry technology transfers in the case of cumulative innovation processes. They may facilitate outsourcing to independent parties as they represent codified information which may be transferred more easily (Arora and Merges, 2004).

Consequently, it is perhaps not surprising that the stock of software-related patents has been growing. In 2004, 4 695 software-related patents were issued to inventors in the US, more than the combined number for the rest the world (2 811). The average annual growth in software-related patenting between 1988 and 2004 was also greater in the US than in all other G7 economies: such patenting by US inventors grew at an average annual rate of 19.5%, compared to 16.1% in Japan and 18.0% in other G7 economies.

Looking at the leading recipients of US patents by country of inventor, Arora *et al*. (2007, Table 7) note that US firms (such as IBM and Intel) are the leading recipients of US software patents in China, India, Ireland and Israel, whereas in Germany, Japan and Korea, domestic firms are the main recipients (all domestic firms in the case of Japan and Korea). Overall, the US accounts for most of US software patents, followed by Japan, Germany and the United Kingdom. More generally, patent data for the US and Europe suggest that inventive activity in software development is concentrated in the US and dominated by US-owned firms. Although there is some evidence that some inventive activity by US firms has started to shift abroad, at present the shift is small and this remains a small share of US firms' overall inventive activity.

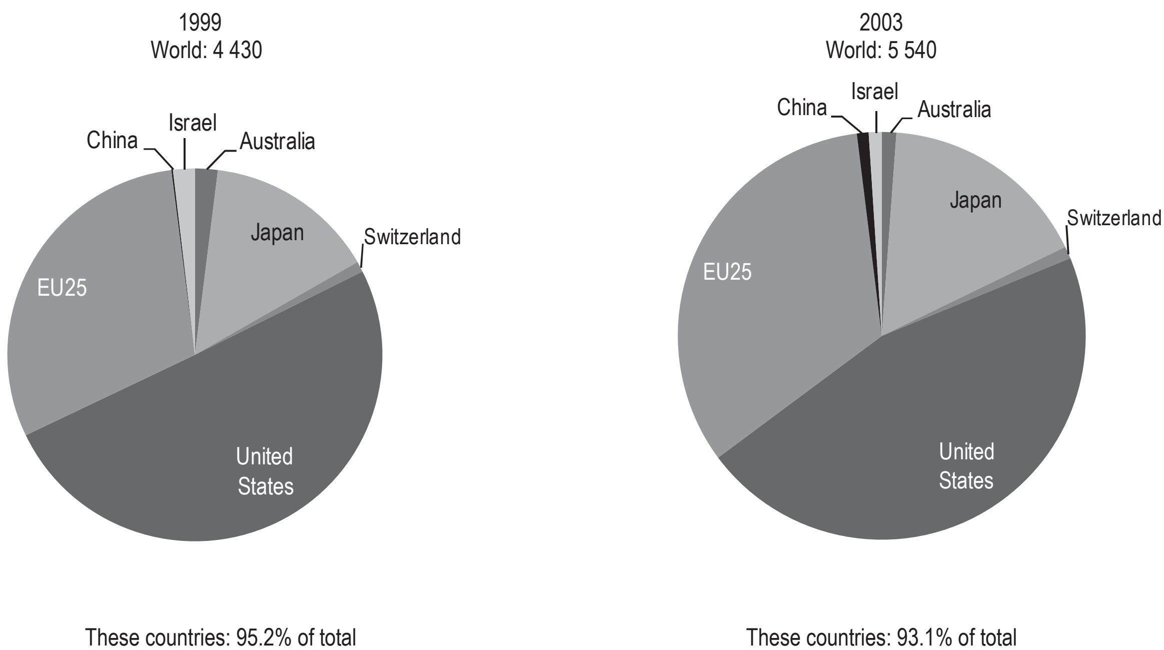
Data from the OECD Patent Database provide further insights on the trends and patterns in patenting. In many countries the share of software-related patents in ICT patents and in total patents increased between 1999 and 2003 (Figure 2.6), although Australia and Israel saw a decline in the share. In China the share of software patents in ICT patents declined while increasing as a share of total patents. The US largely dominates the worldwide distribution of software-related patents, although its share decreased slightly between 1999 and 2003 (Figure 2.7). The EU25 countries collectively place second, while Japan accounts for the second largest individual country share.

Figure 2.6. The share of software-related patents in ICT and total patents, 1999 and 2003



Source: OECD Patent Database.

Figure 2.7. Share of world software-related patents, 1999 and 2003



Source: OECD Patent Database.

The bulk of software-related patents are filed by firms whose primary activity is not software. There is no official definition of a software-related patent though as it is not recognised as such in the international patent classification system. Analysts have adopted several methods to try to identify software patents. One is to identify software-related patents as those that fall in a particular International Patent Classification (IPC) class, subclass or group (*e.g.* Graham and Mowery, 2003). Another approach has been to search for keywords. Thus, Bessen and Hunt (2004a, 2004b), for example, defined software patents as those that include the words "software" or "computer" and "program" in the patent document description.[23] Hall and MacGarvie (2006) propose a definition that combines the two previous methods with one constructed on the basis of the patent classes and subclasses that contains patents assigned to fifteen of the largest software firms. A recent study for Europe, based on a combination of these three methodologies, shows that the software sector accounted for only 1.3% of software-related patents. The sector "electronic instruments and telecommunications equipment" alone accounted for 61.9% of software patents, followed by "telecommunications" with 8.2% and "motor vehicles" 3.9%.

Bessen and Hunt (2004a, 2004b) in a study for the United States found that between 1994 and 1997, software-related patents tended to be US owned (70%), by firms (88%) rather than government (2%), and to have a US inventor (69%). After the US, the top countries are Japan (18%), Germany (3%), the UK (2%) and Canada (2%). Some 75% of software-related patents in their sample were obtained by manufacturing firms, especially in the electronics and machinery industries, which include computers. Only 5% were acquired by software publishers (SIC 7372); other software service firms, excluding IBM, accounted for 2%. IBM alone accounted for 13% of software-related patents in the sample, being consistently the largest software patentee.[24]

The market environment for R&D

The market environment for R&D and innovation in the software sector is increasingly characterised by the globalisation of software activities. Rapid developments in ICTs and the roll-out of high-speed broadband Internet and communications networks enable these changes and create opportunities for new business approaches such as "software as a service" (as discussed in Chapter 1). Globalisation in the sector is partly reflected in increased numbers of firms and communities (*e.g.* universities, institutes, governments) collaborating across national borders, but also increased trade in software goods and services, international movement of software engineers and other IT specialists, offshoring of software activities, and expansion of the international operations of large multinational enterprises (MNEs).

Business sector R&D, in general, is rapidly internationalising. This is largely driven by the changing strategies of MNEs, which account for the bulk of global business R&D. In the past, most MNEs had the tendency to keep R&D "at home", where the headquarters were located, while globalising other activities. Now, they increasingly are adopting strategies of global technology sourcing. Software sector firms are increasingly situating their R&D activities strategically at various locations around the world, tapping into regional strengths or positioning themselves in key markets. This can be in the form of research laboratories or, for example, on-site with other collaborators such as clients. Indeed, development for some types of software requires physical proximity to the client.

International competitiveness

A recent report by the Economist Intelligence Unit (EIU, 2007) reviewed the competitiveness of key countries with respect to the information technology sector (Table 2.4). The report identified the United States as the top location for firms in the sector, with particularly high scores for availability of highly skilled ICT-related human capital and a sound legal environment. The attractiveness of the US market is illustrated by the fact that such a large portion of leading software firms are headquartered there (as discussed above). Many of the factors that drive a country's attractiveness as a location for IT investment have also been identified as enabling and enhancing innovation (*e.g.* Jaumotte and Pain, 2005a,b,c).

Table 2.4. IT Industry competitiveness index 2007, selected countries

	Overall index score	Business environment[1]	IT infrastructure[2]	Human capital[3]	Legal environment[4]	R&D environment[5]	Support for IT industry development[6]
Category weight:		**10%**	**20%**	**20%**	**10%**	**25%**	**15%**
United States	77.4	97	81.3	96.4	92	39.8	86.8
Japan	72.7	82	52.3	67.4	79	84.3	77.1
Korea	67.2	80	61.7	74.8	66	56.6	74.3
United Kingdom	67.1	95	69.4	81.6	88.5	23.2	84.9
Australia	66.5	92	75.9	76.2	87	21.1	86.2
Canada	64.6	88	87.5	65.9	82	15.5	86.8
Netherlands	62.9	91	72.4	59.1	87	23.5	86.1
Germany	58.2	88	58	59.4	85	28.9	68
France	55.8	83	54.3	60.3	83.5	20.6	73.6
Israel	54.5	83	45.8	64.8	75.5	24.9	68.8
Spain	46.1	80	29.6	61	78	6.6	70.1
South Africa	33.4	77	8.9	40.8	63.5	1.5	60.6
India	29.1	60	0.5	49.6	48	0.7	54
Russian Fed.	28	48	8.6	56.8	38.5	6.3	31.5
China	27.9	47	8	44.7	49	2.2	48.1

1. Business environment includes government policy towards foreign capital, degree of private property rights protection and freedom to compete.

2. IT infrastructure covers market spending on hardware, software and services, desktop per 100 people, broadband connections etc.

3. Human capital includes enrolment in tertiary level science programs and employment in technology sector.

4. Legal environment measures the comprehensiveness, transparency and enforcement of IP legislation, together with the status of national cybercrime laws.

5. R&D environment includes gross government and private sector expenditure on R&D, number of new patents registered and receipts from royalty and license fees.

6. Support for IT industry development covers access to medium term finance for investment and government policies.

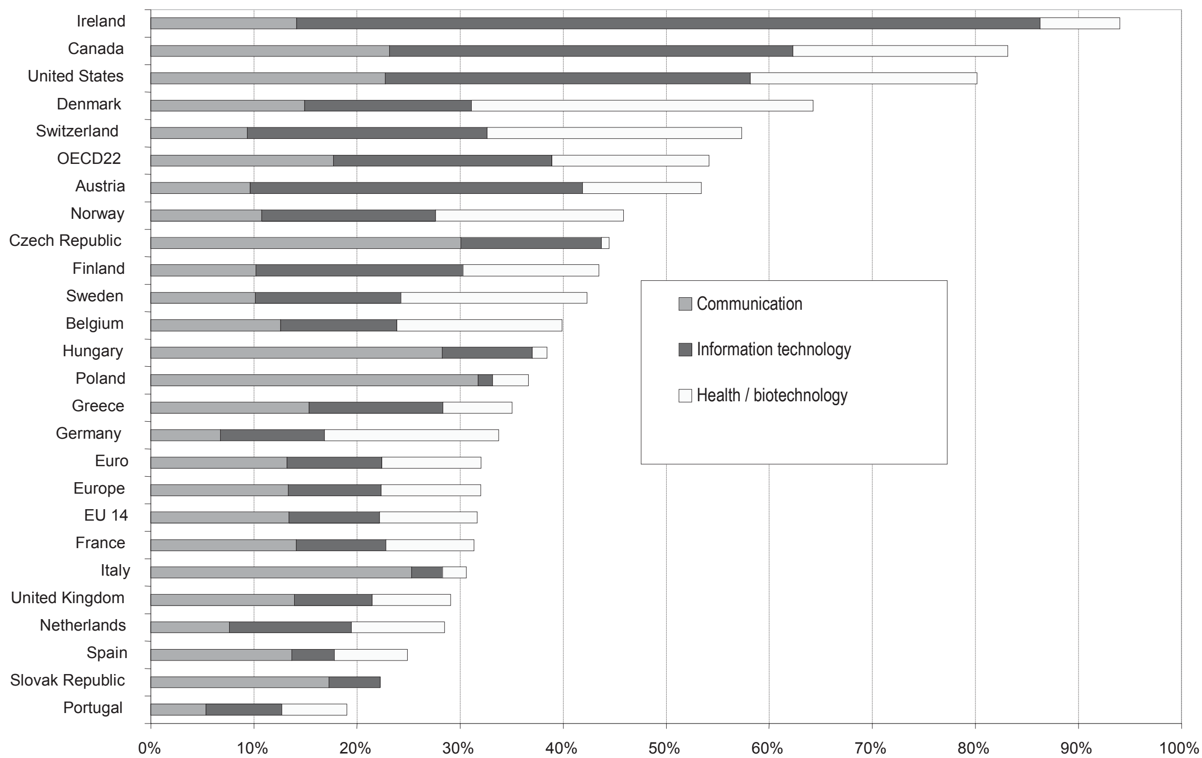
Source: EIU (2007), "The Means to Compete: Benchmarking IT Industry Competitiveness", Economist Intelligence Unit.

Japan and Korea ranked second and third, due in part to strong scores with respect to the environment for R&D according to the EIU definition (based on R&D expenditure, patenting and revenues from royalties and license fees). There are five European Union countries among the top 15 locations. A number of emerging countries, including South Africa, India, Russia and China also figured in the listing, although lagging behind the other countries by a fairly wide gap. India and China have exhibited strong improvements ICT sector performance in recent years, fostered by favourable factors such as relatively low wages and large workforces with growing pools of skilled ICT professionals.[25] Relative to OECD economies, they still have some distance to go in order to catch up on factors such as ICT infrastructure, skills, and the legal environment to further improve overall ICT sector competitiveness and increase their attractiveness for investment in software including R&D activities (van Welsum and Xu, 2007).

Among the various elements influencing ICT competitiveness, the importance of human capital to successful innovation in the software sector is frequently underscored by stakeholders and analysts, including with respect to R&D. This is true more broadly in the economy, as can be seen in the evidence from macroeconomic studies on the determinants of innovation and the outcomes of various innovation studies. But it is particularly true with respect to the software sector, which remains particularly human capital-intensive despite substantial technical progress in such areas as software engineering and development tools. The responses to an OECD Business Questionnaire addressed to software developers also highlighted the importance of human capital (OECD, 2008a).[26]

Figure 2.8. Share of high-technology sectors in total venture capital

As a percentage of total venture capital investment, 2000-03



Note: Total venture capital investment consists of early and later stage venture capital as well as, except for the United States, buyouts. European Union 14 comprises data from Austria, Belgium, Finland, France, Germany, Greece, Ireland, Italy, Netherlands, Portugal, Spain, Denmark, Sweden, and the United Kingdom. OECD22 comprises data from European Union 14 countries as well as from Norway, Switzerland, the Czech Republic, Hungary, Poland, the Slovak Republic, Canada and the United States.

Source: Lumpkin and Thompson (2006), "The SME Financing Gap: Theory and Evidence", OECD Working Paper DAF/CMF (2006), Directorate for Financial and Enterprise Affairs, OECD, Paris, based on data from EVCA (Europe); NVCA (United States); CVCA (Canada).

Access to finance, and venture capital in particular, is very important for highly innovative and rapidly evolving sectors such as the software sector, especially since the development of software is costly and time consuming. Small software firms in particular tend to be venture-backed (Mann, 2005). The share of high-technology sectors in total venture capital (Figure 2.8) is indicative of the innovative potential generated by venture capital. According to Lumpkin and Thompson (2006), just over half of venture capital investment goes into high-technology firms in OECD22[27] economies, but there are large disparities among countries. The share is highest in Ireland, Canada, the United States and Denmark, which all score above the OECD22 average. The IT sector accounts for the largest share in several countries, including Canada, Ireland and the United States. The high-technology share is also relatively high in some smaller countries which have relatively low overall levels of venture capital to GDP, such as some of the East European countries, but there the share is largely dominated by the communications sector.

The United States stands out in terms of inputs of financial capital to emerging software firms, although the amount has at times fluctuated. According to a Price Water-house Coopers report (2007), overall venture capital in software in the United States came to over USD 5.1 billion in 2006 and close to USD 4.9 billion in 2005. The National Venture Capital Association estimates venture capital software investments in the United States during 2007 at USD 5.4 billion (NVCA, 2008). Globally among ICT industries, the software industry received the most venture capital funding annually throughout the 2000-2006 period. Using a more narrowly specified methodology, one comparison by Arora *et al.* (2007) of *disclosed* rounds of venture capital financing for software products and services confirmed the United States to be a leader in this form of financing. In 2005, for example, the amount of such financing in the United States came to USD 138 million while the amount in the remaining G7 countries combined came to around USD 2 million and also to about USD 2 million in a combined country group of emerging software innovators including India, Ireland, Israel, Brazil and China.[28]

Locational advantages

In the context of ICT competitiveness and globalisation of the market environment for innovation, two factors that are sometimes raised as being of particular interest include: wage differentials and infrastructure. While relevant, both factors have limitations in terms of promotion of investment in R&D.

Table 2.5. Average salaries of software programmers

Country	Salary range (USD)
Poland and Hungary	4 800 – 8 000
India	5 880 – 11 000
Philippines	6 564
Malaysia	7 200
Russian Federation	5 000 – 7 500
China	8 952
Canada	28 174
Ireland	23 000 – 34 000
Israel	15 000 – 38 000
United States	60 000 – 80 000

Sources: van Welsum and Xu (2007) “Is China the New Centre for the Offshoring of IT and ICT-enabled Services?, DSTI/ICCP/IE(2006)10/FINAL, Directorate for Science, Technology and Industry, OECD, Paris, available at *www.oecd.org/sti/offshoring*; Bardhan and Kroll (2003), “The New Wave of Outsourcing”, University of California Berkeley, Fisher Centre for Real Estate and Urban Economics, Fisher Centre Research Report No. 1103.

Wages

Wage differentials are often mentioned as a driver in the locational decisions of firms. Table 2.5 presents the example of average salaries for software programmers across selected countries, highlighting the large differences. To some extent, wage differentials reflect differences in productivity. However, with regard to R&D activities (and many other aspects of business), simple wage differentials – even normalised for productivity differences – may not be enough to drive such locational decisions. In sizing up availability of human resources for R&D, other quality factors such as education, skills, language, experience and talent also count as they may influence the creative processes.[29] As noted by respondents to the OECD business questionnaire (OECD, 2008a), the availability of trained human capital is a critical input to the software development process.

ICT infrastructure

The availability and quality of basic ICT-related infrastructure are also important for determining the location of globalised software activities. The quantity and quality of infrastructure and their prices vary greatly across countries. Some countries have large absolute amounts of infrastructure, which is one indication of national capacity for receiving ICT-enabled offshored software and R&D activities. For example, China has more PCs than Germany and more Internet subscribers than the United States. Brazil, India and Russia each have about as many PCs as Canada or Italy, and Brazil and India have slightly fewer Internet subscribers than Canada. However, apart from China, these economies' broadband subscriber numbers are much lower, and broadband costs in all of them are much higher than in most OECD countries.[30] Respondents to the OECD Business Survey (OECD 2008a) ranked infrastructure as being of medium importance for the software development process, suggesting that it may be a necessary but not sufficient condition for locational decisions. This may be particularly true in the specific case of R&D, which may have high technological requirements, but which may also require the availability of high calibre human capital in order to deliver expected results.

Globalisation: shaping the market environment for innovation in the software sector

Globalisation now plays a key role in shaping the market environment for innovation in the software sector, by expanding the scale of the market for software products and by expanding the options for enterprises in term of their structure and inputs including in relation to R&D. Globalisation of software activities contributes to increased access to foreign knowledge and inventions, which has been identified as an important driver of innovation at the aggregate level (Jaumotte and Pain, 2005a,b,c, d).

While there are many manifestations of globalisation in the sector, this section focuses on selected aspects in order to provide an illustration of developments in relation to the market environment and R&D. The section begins with consideration of the extent of software sector globalisation in light of trade developments. It then moves to consider the role of MNEs, which play a key role in relation to globalised R&D processes in the sector. Also, the role of outsourcing is then considered. Next, locational advantages are addressed in terms of attractiveness for investment. The penultimate section considers the relationship of the software sector to R&D in other sectors in a globalised market. Finally, the section briefly returns to considers open innovation (see discussion in Chapter 1 as well) in the context of globalisation.

Box 2.1. The challenges of measuring software trade

There are many measurement issues associated with software trade. First, as border valuations are based on physical media (diskettes, CD-ROMs), rather than on content (the software), the value of the software traded is likely to be significantly understated. Second, the bundling of software with hardware leads to significant mis-measurement (equipment trade is overstated and software trade is understated). Third, trade statistics do not measure the value of copyright works sold in foreign markets. This is referred to as the so-called "gold master" problem in cases where only the original software product is counted in international trade, but it is then copied many times for sale in the importing country. Fourth, trade statistics do not measure the value of software transmitted electronically across borders, which accounts for a rapidly increasing share of sales, or the rise of application service providers of software (ASPs). Many of these issues, such as the treatment of software in the Balance of Payments and in National Accounts (*e.g.* through royalty payments for use of intellectual property), are receiving greater scrutiny as the value of these payments increases rapidly.

Some countries, like the United States, publish more detailed software trade data. For example, the US Bureau of Economic Analysis publishes general-use software trade data (*e.g.* data on software publishing sales to foreign persons by US MNEs), although they do not include affiliated transactions. The US Census Bureau publishes data for packaged software shipped on media, but the data may include the value of other intellectual content (*e.g.* movies) on the media in certain cases.

Source: OECD (2006), *OECD Information Technology Outlook 2006,* OECD, Paris.

Trade in software goods and services

Software goods

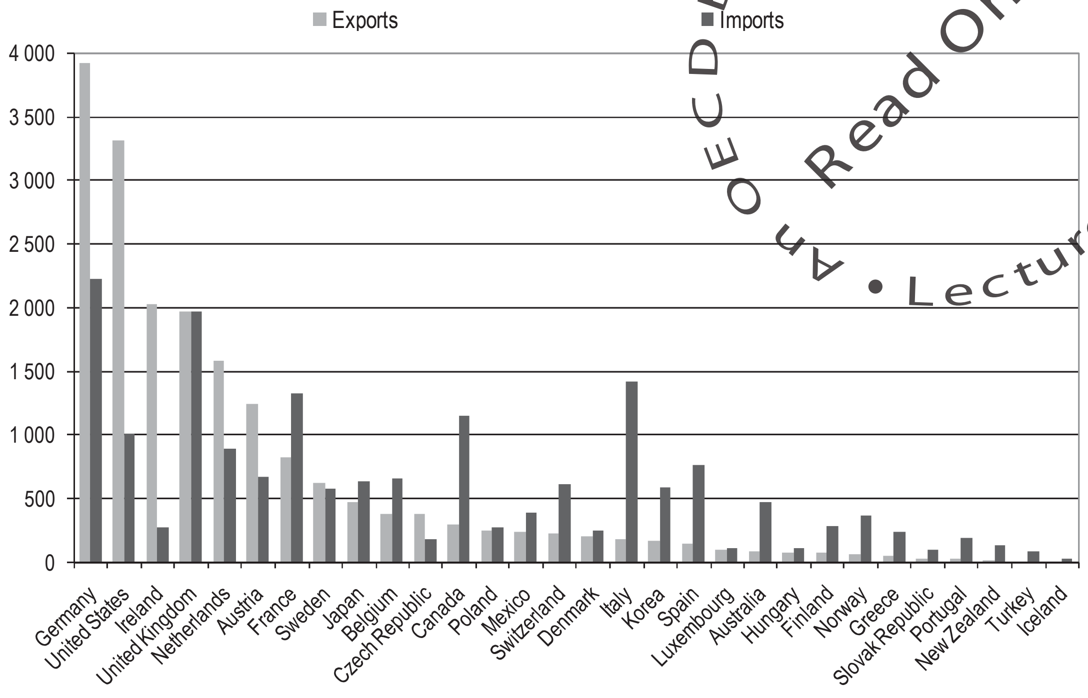
Software cannot be directly identified in trade data (Box 2.1), but one estimate has put the total OECD exports of software goods at about USD 19 billion in 2006 and imports around USD 18 billion (OECD, 2008c). Between 1996 and 2006, exports increased by 5.2% a year, while imports increased by 5.6% a year. Figure 2.9 shows data for 2006 for selected OECD countries. In that year, Germany and the United States were the leading exporters, with exports of over USD 3 billion each, while Germany and the United Kingdom were the leading importers with imports of over USD 1.5 billion each. Italy, Canada, Spain and France experienced the largest trade deficits in software goods, while the United States, Ireland and Germany recorded the largest surpluses.[31]

Computer and information services

Data on trade in software services are not widely available, so the category "computer and information services" is used as a proxy.[32] Between 1996 and 2006, OECD exports of computer and information services increased on average by 20% a year, from around USD 13 billion to USD 86 billion. OECD imports increased by 15% a year from USD 12 billion to USD 54 billion (OECD, 2008c). In 2006, Ireland, the United Kingdom and the United States were the leading exporters, with volumes of over USD 10 billion each (Figure 2.10). Germany and the United States were the biggest importers, with volumes of about USD 7 billion or more. While Ireland and the United Kingdom experienced relatively large surpluses in trade in computer and information services, some other countries, *e.g.* Japan and Italy, saw a trade deficit.

Figure 2.9. OECD software goods trade, 2006

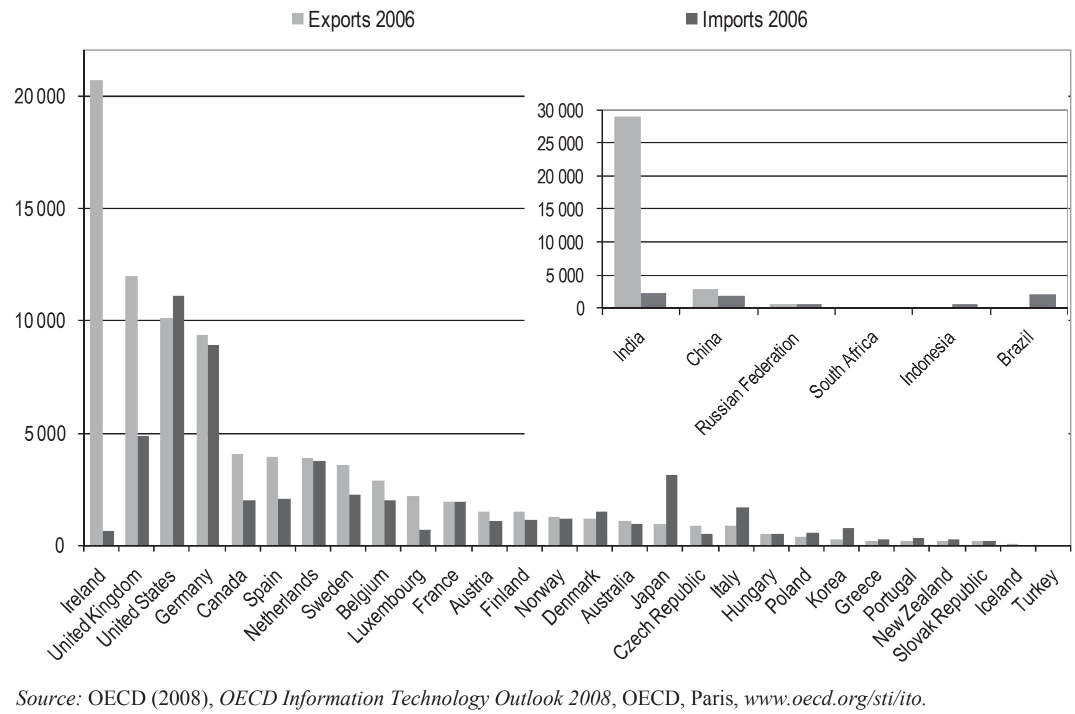
USD millions



Source: OECD (2008), *OECD Information Technology Outlook 2008*, OECD, Paris, *www.oecd.org/sti/ito.*

Figure 2.10. Computer and information services trade, 2006

USD millions



Source: OECD (2008), *OECD Information Technology Outlook 2008*, OECD, Paris, *www.oecd.org/sti/ito.*

The case of Ireland as the largest exporter merits further explanation. The large export volume and large surplus on trade in computer and information services is partly a product of statistical measurement and tax issues. As for statistics, the volumes for Ireland are somewhat inflated because the data include software license fees in computer and information services, while other countries record them separately under "royalties and license fees" (OECD, 2006a). Nevertheless, taking into account computer and information services, software goods and software-related royalties and license fees, it can be said that Ireland is a major producer and exporter of software and IT services. As for taxes, Ireland has established an advantageous tax regime and a number of software firms use Ireland as a location to set up export operations, in part to benefit from its provisions (OECD, 2006a).

The role of multinationals in globalisation of software and R&D

In the context of globalisation, MNEs provide a natural mechanism to foster, select and co-ordinate R&D projects and activities (OECD, 2008b). Firms rely on specific competencies, learning processes, and communication systems that reduce the cost of co-ordinating different individuals and parts of the organisation. They can develop competitive advantage via operation of effective learning processes, capabilities to co-ordinate and integrate internal activities, and the ability to modify strategies and competencies according to the changing market environment (Teece *et al.,* 1997).

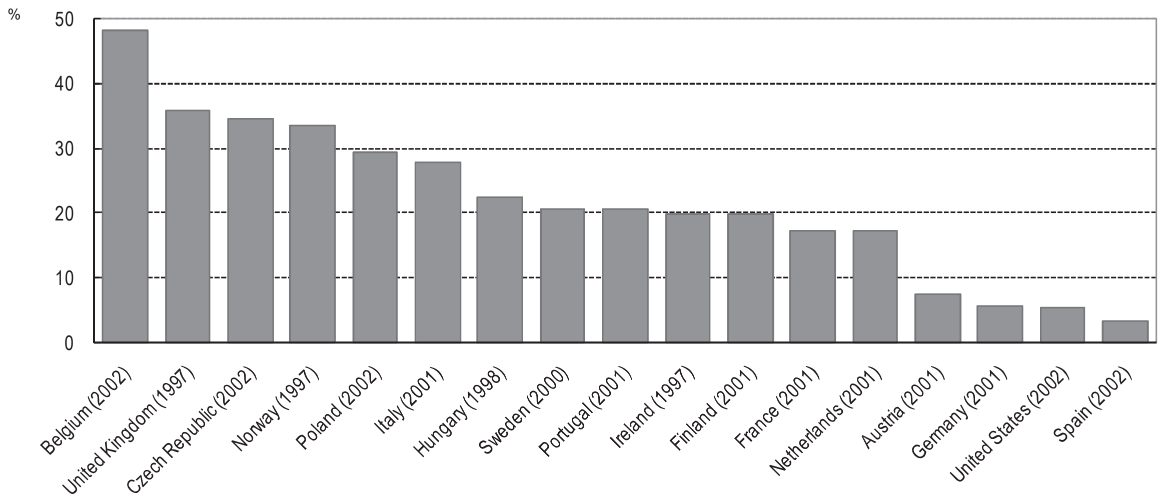
Business sector R&D is rapidly internationalising, largely driven by the changing strategies of MNEs which account for the bulk of global business R&D. While in the past MNEs tended to keep core R&D in their home countries, now these firms are increasingly shifting to a mixed strategy of global technology sourcing (OECD, 2008b). Nonetheless, proximity continues to play a role in part because relay of tacit knowledge tends to rely on proximity, practice and learning-by-doing. By locating close to each other, people can access information, monitor other people's behaviour, and foster communication among individuals, thereby reducing the complexity and uncertainty of the innovation process. At the same time technological developments in ICTs are enabling a broader and faster exchange of codified knowledge and ideas across international or even global networks of collaborators.

In the case of software, international expansion is one of the priorities of top software MNEs, often as a complement to other dimensions of growth strategies including such elements as internal product development and joint research with domestic institutions. Recent field studies have suggested that these firms now undertake substantial software R&D outside of their home countries. Many of the top software firms are establishing R&D centres in an increasing number of locations, including India, Ireland, and Israel (see case studies of these countries at the end of this chapter). In other cases, they are partnering with foreign firms including small and medium size enterprises. Furthermore, whereas R&D undertaken abroad used to focus essentially on adapting products to the local market, these days foreign R&D goes much beyond that and also includes original and innovative development work, in some cases attempting to tap into local knowledge and sources of new technologies.

There many ways in which MNEs can expand their R&D operations abroad, including through affiliates, acquisitions and collaborative or partnership agreements with local companies. In many countries, foreign affiliates play an important role in total national R&D. In most OECD countries, the share of affiliates under foreign control in total business sector R&D expenditures increased between 1995 and 2003, except in

Spain and Turkey. In Ireland in 2003, more than 70% of total business sector R&D expenditure is attributed to foreign affiliates. The share of foreign affiliates in total business sector R&D expenditures is lowest in Japan where business sector R&D is dominated by domestic firms. Comparable figures are not available for the software sector, but an indication of the extent of foreign affiliate presence is available from an OECD database on foreign affiliates (FATS). Figure 2.11 presents the share of foreign affiliates in the revenues of the Computer and Related Services sector for selected OECD countries as of 2002. The data confirm the active engagement of foreign affiliates in the sector in these OECD countries, with shares ranging from just a few percent in Spain to nearly 50%in Belgium.

Figure 2.11. The share of foreign-controlled affiliates in revenues of computer and related services enterprises (ISIC 72)



Source: OECD (2005), OECD Statistics on Measuring Globalisation, Volume II: Services (FATS), April.

Trends in software offshoring

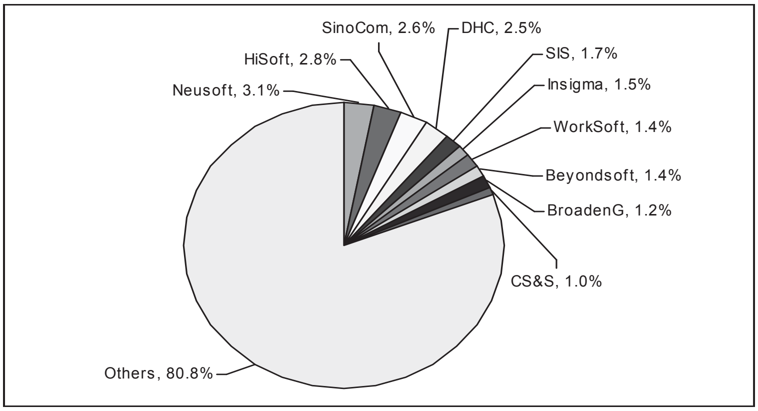
Rapid advances in ICTs are increasing the tradability of many business services, including software services, and are also creating new tradable services. As services become more tradable and increasingly independent of location, firms are starting to offshore certain business functions, such as administrative support units, IT-related activities, and research and consultancy services. Often, their objective is to refocus their operations on their core activities and thereby increase their competitive advantage. Attractive destinations include countries with relatively lower labour costs and a talented workforce. Due to the wage-cost advantage and the large pool of English-speaking skilled labour, India has become a prime location for ICT-enabled-services offshoring in recent years (OECD, 2006a). More recently, ICT-enabled offshoring of services has also moved into China (van Welsum and Xu, 2007). Offshoring of services increases access to foreign knowledge and talent, thereby potentially also stimulating and facilitating innovation.

Box 2.2. Software services outsourcing in China

China's software services outsourcing market amounted to around USD 0.3 billion by the first quarter of 2006. It is growing rapidly, but from a relatively small base. Neusoft, Hisoft and SinoCom are the top three Chinese software services outsourcing companies. The total market share of the top ten software offshoring and outsourcing companies is only 25%, and the Chinese software outsourcing industry is highly fragmented and lacks large firms dedicated to outsourcing (IFC, 2005). The largest Chinese software firms often engage in a wide variety of other business activities in addition to supplying software outsourcing services, such as software development and hardware distribution.

Despite rapid growth of Chinese software services suppliers, they remain small relative to the top Indian firms. The top three Chinese software firms had combined revenues of around USD 500 million in 2005 from all activities, with around one-fifth coming from providing outsourced services, some of which are offshored outsourcing services to foreign customers. This compares with USD 5.5 billion for the three top Indian software and IT consulting firms (Tata Consultancy Services, Wipro and Infosys) the major share from providing outsourced services, giving a good indication of the relative strengths of ICT services firms from the two countries. The large Indian firms are reaping economies of scale and establishing global strategies to both export their services and locate close to their customers in their main markets.

Supplier shares of the software outsourcing market in China



Source: Analysys International (2006).

By sector of demand, high-technology products, consumer electronics, telecommunications and finance account for 95% of the software outsourcing market. High-technology is the most important, accounting for 60% of the market for software outsourcing. The next is consumer electronics, accounting for 25%. Most software outsourcing services go to Japan (59% in the first quarter of 2006). The US and Europe account for 23% and Hong Kong (China) for 11%. Japan has also moved some of its software development work to China. China currently has six software-export bases: Beijing, Shanghai and Tianjing — the largest municipalities in China; Dalian (in Liaoning Province), Shenzhen (in Guangdong Province), and Xi'an (in Shaanxi Province).

Source: van Welsum and Xu (2007), "Is China the New Centre for the Offshoring of IT and ICT-enabled Services?," DSTI/ICCP/IE(2006)10/FINAL, Directorate for Science, Technology and Industry, OECD, Paris.

Early evidence from these developments suggests that it was mainly the lower end of software services and production that were offshored rather than the core activities (Arora *et al.*, 2001). Thus, US firms were found not to offshore or outsource activities such as requirement analysis, specification, and high-level design, or larger scale system integration types of activities to India, although the leading Indian software firms had the ability to provide these high-end services. In fact, the need for proximity might hamper the offshoring of more highly value-added activities such as software design and R&D.

The impact of offshoring in the software sector is not clear cut. Some argue that it causes tensions in job market in the parent country of software MNEs, due to the perceived substitution effect of software professionals. Others, such as Arora *et al.* (2001) argue that the Indian software industry is largely complementary to the US software industry. Indian firms provide essential maintenance and development services, enabling US firms to use their scarce in-house ICT staff for higher value added work, such as design and develop new types of applications. Indian firms often act as sub-contractors to established US software services firms and systems integrators. In addition, many of these US firms rely on Indian programmers and have significant India based operations. More recently however, Indian firms have started to move up the value chain and perform more highly skilled activities with greater value-added (OECD, 2006a). This potential for moving up the value chain has also been analysed for other emerging countries, such as China (*e.g.* van Welsum and Xu, 2007, see also Box 2.2). At the same time, there may be new opportunities for niches of low cost domestic sourcing of certain activities (ITAA, 2007).

Collaboration and open innovation in R&D

As the complexity and range of application for software has grown, software developers have moved to expand collaboration as means of increasing the base of resources involved in R&D processes. Such expanded collaboration can take many different forms ranging from a closed initiative among two partners to an open project accessible by thousands. This section first considers evidence from innovation surveys in the European Union and Japan, to examine the nature of collaboration in two OECD economies.[33] The section then moves to consider open innovation more specifically, followed by an examination of the linkages between the traditional software sector and software development in other sectors.

Innovation surveys

Indicators on some aspects of innovation can be obtained from so-called innovation surveys, although typically not at the level of the software sector. These data are available for the EU and several non-EU European countries including Japan, although they are not necessarily strictly comparable on an international basis.

For the European Union, data from the Community Innovation Survey (CIS)[34] provide some insights into R&D for business services, but the software sector cannot be separately identified within this category.[35] Among the various sectors between 1998 and 2000, the business services sector showed the largest share of enterprises with innovative activity (64%),[36] followed by financial intermediation (58%) and manufacturing (47%). Furthermore, 61% of success innovators in business services were both product and process innovators, while the rest focussed on either product or process innovation. Thus, the larger sector around software tends to be relatively innovative by this indicator.

Many enterprises with innovation activity also reported undertaking other strategic and organisational changes, and notably more than enterprises without innovation activity: strategy (67% vs. 30%), management (56% vs. 20%), organisation (71% vs. 37%), marketing (59% vs. 22%) and aesthetic or other subjective changes (54% vs. 16%). The importance of these changes and innovations is double fold as they not only reflect those made by firms in the software sector, but also those enabled by software in firms throughout the economy (*e.g.* new business methods, management practices, flexible working arrangements etc.).[37]

The CIS provides some interesting insights into the role of collaboration for business service sector firms in Europe. The most important sources of information for innovation as cited by firms in the sector included: sources from within the enterprise (48%), clients or customers (40%), professional conferences, meetings, journals (24%), suppliers of equipment, materials, components or software (18%), competitors and other enterprises from the same industry (15%), fairs and exhibitions (14%), universities or other higher education institutes (12%), other enterprises within the same industry group (9%) and government or private non-profit research institutes (6%). Some 31% of business services sector firms with innovation activity reported having had access to public funding (any source), 17% from local or regional authorities, 16% from central government, 9% from the EU, and 8% from the EU's 4th (1994-1998) or 5th (1998-2000) Framework Programmes for RDT. Furthermore, 34% of business services sector firms with innovation activity were involved in innovation co-operation: 31% with national partners, 10% with EU/EFTA partners, and 1% with candidate countries. The largest reported innovation impacts reported by these firms were product oriented impacts (increased range of goods or services: 42%, improved quality in goods or services: 44%, and increased market or market share: 32%). Other effects included meeting regulations or standards (16%), increased production capacity (15%) and improved production flexibility (13%). Thus, these results confirm the heavy engagement of a range of innovation inputs beyond the walls of the firms in the larger sector surrounding software.

Problems and barriers to innovation were also reported among these firms in the European Union. Some 55% of business services firms with innovation activity reported that their innovation activity had been seriously delayed, 29% that it had been prevented from being started, and 24% faced other serious problems, including: too high costs (31%), lack of appropriate sources of finance (30%), excessive perceived economic risks (24%), lack of qualified personnel (24%), and insufficient flexibility of regulations or standards (13%). For those without innovation activity, the reported barriers included too high costs (18%), lack of appropriate sources of finance (18%) and perceived economic risks (14%).

The Japanese National Innovation Survey (NISTP, 2003) distinguishes the sector "computer and related services", and its sub-category "software consultancy and supply". It finds that for the period 1999-2001, 40% and 43% of firms, respectively, were innovators, the proportion rising with firm size (as much as 60% and 68% of large firms, respectively). Furthermore, 88% and 91% of innovating firms in these sectors were identified as product innovators, and 48% and 47% of the firms as process innovators. Innovators with new-to-market products accounted for 40% and 42% of innovating firms.

In software consultancy and supply, all innovating firms in the Japanese survey reported the implementation of management strategies, 90% knowledge management, 83% organisational changes, 11% marketing changes and 36% aesthetic changes; 64% reported conducting intra-mural R&D. The reported impacts of innovation included:

improved quality of goods and services (41%), increased goods and services (38%), expanded market or increased market share (21%), improved production flexibility (17%), reduced materials and energy usage per production unit (15%), increased production capacity (14%), reduced labour cost per production unit (9%), satisfied regulations or standards (6%), and improved environmental impact or health and safety aspects (2%).

With respect to financial support from local or regional public authorities and central government in Japan, respectively, 8% and 4% of innovating firms in software consultancy and supply reported having received a grant or a subsidy, and 7% and 3% a loan or a credit guarantee, respectively. Partners in co-operation for innovation included: other enterprises within the industry group (18%), universities or other higher education institutes (16%), commercial laboratories, R&D enterprises, and suppliers of R&D support services (12%), government or private non-profit research institutes (10%), and consultants (10%).

In the Japanese survey, innovating firms in the computer and related services cited the following sources for important innovation and new innovation project ideas: the R&D department within the enterprise (51%), clients (40%), the marketing department within the enterprise (30%), professional journals or academic journals (29%), suppliers of equipment, materials, components or software (28%), the production, manufacture or maintenance department within the enterprise (26%), trade fairs or exhibitions (17%), professional conferences or meetings (16%), competitors and other enterprises from the same industry (15%), consultants (12%), universities or higher education institutes (10%), government or private non-profit research institutes (6%), and commercial laboratories, R&D enterprises and suppliers of R&D support services (5%). Thus, in this software-related sector in Japan, innovators were also heavily engaging resources from beyond the walls of the firm in the innovation process.

Problems and barriers to innovation were also reported in the Japanese survey. Thus, 35% of innovating firms in computer and related services reported that their innovation activity had been seriously delayed, 16% that it had been prevented from being started, and 58% faced other serious problems, including: excessive economic risks (27%), too high innovation costs (25%), lack of information on markets (25%), lack of appropriate financial sources (22%), lack of information on technology (24%), lack of qualified personnel (22%), organisational rigidity within the enterprise (12%), lack of customer responsiveness to new goods or services (10%), and insufficient flexibility of regulations or standards (4%).

Collaborative approaches and small and medium size enterprises (SMEs)

High-growth SMEs tend to be very market-oriented and respond to market changes with product innovations, often also closely related to process innovations (OECD, 2008b). They tend to aim for improved product quality and customer satisfaction rather than reduced costs. This is an example of how firms can create value from their intellectual assets. Most high-growth firms relied on networking and public-private relationships to develop innovative products and processes, and often operating without their own R&D department.

SMEs may strive to adopt collaborative approaches to invention when they do not hold sufficient internal competences and resources to develop an invention autonomously. Thoma (2008) looks at copatenting (across all sectors) to examine the extent of R&D

collaboration among firms. Although copatenting is not a common form of collaboration, he finds that monoregional firms, occasional innovators, and firms from smaller countries tend to use copatenting relatively more than multiregional firms and serial innovators. It may be that these firms, employ such approaches to leverage their strengths, while engaging partners with complementary strengths in order to compete in the market place.

In the specific case of software SMEs, there seems to be an expansion of such activity. For example, Athreye (2005) and Arora and Surendrakumar (2006) find that India is experiencing rapid growth of R&D in small-sized companies, particularly in embedded systems. A portion of such growth is fuelled through collaboration with foreign partners.

Open source approaches

Open source software development contributes to the growth and structural change in the software sector. An indication of that change is the growth of the mixed source or hybrid computing models. One of the guiding principles for open source software is that, by sharing source code, developers co-operate under a model of peer-review and take advantage of "parallel debugging" that may lead to innovation and rapid advancement in developing and evolving software products (Dempsey *et al.*, 1999). Other guiding principles are licensing models offering innovators and users differing levels of access (or non-access) to and permissions regarding source code and a range of business models for commercialising open source software. Broadband Internet connectivity has enabled and reinforced the open-source notion of co-operative, peer-reviewed software development that can be deployed on a global scale (Box 2.3). This model is inherently based on combining diverse sources of knowledge and talent and in this manner aims to be conducive to innovation.

Box 2.3. Open source software: an example of global co-operative and innovative networks

Software development projects illustrate how applied ICT tools and broadband can serve to facilitate innovation. With broadband-enabled infrastructures, participants in such projects can be located in any geographical location. Free and open source software projects are one relatively well-developed form of internet-based innovation community, where innovations are disclosed freely. Some open source software is created in communities rather than firms, and enhancements to the codes are available to everyone on an equal basis. It is a collaborative, community model based on a process that does not allow any contributor to exert a proprietary claim to intellectual property on any portion of the code being developed within the open source framework (OECD, 2007a). However, there are ample business opportunities surrounding this development, for example with respect to add-on software, services, consultancy and training.

Necessary (although not OSS-exclusive) tools and infrastructure available to participants in an open source project include email lists for specialised purposes that are open to all and are publicly archived. Programmers contributing to open source software projects tend to have essential tools, such as specific software languages, in common. These are generally not specific to a single project, but are available on the web. Basic toolkits held in common by all contributors also tend to greatly ease interactions. Furthermore, version-control software allows contributors to insert new code contributions into the existing project code base and test them to see if the new code causes malfunctions in existing code. If so, the tool allows easy reversion to the status quo ante. This makes "try it and see" testing much more practical, because much less is at risk if a new contribution inadvertently breaks the code. Toolkits used in open source projects have evolved with practice and are continuously being improved by user-innovators. Individual projects can now start up using standard infrastructure sets offered by sites such as Sourceforge.net.

Sources: von Hippel (2005), *Democratizing Innovation*, The MIT Press, Cambridge, Mass.; and OECD Productivity Database.

One way to examine the geographical distribution of developers of open source software is to look at developers' contributions to source code through their identities in software project data (EC, 2006). The United States, Canada, the European Union, Australia and New Zealand have the highest number of open source participants, and their market concentration of open source software is reportedly high as well (EC, 2006). A large number of both public and private organisations report some use of open source software in most application domains.[38]

Software development and "non-software" firms

The tight links between the software sector and other industries are highlighted in several OECD studies (2002b, 2004a, 2006a, 2006b). These studies point to the relationship of the software sector to computer and communications industries through technological links and through the business activities of firms that are often present in multiple markets. This reflects, in part, the fact that software is produced both as a final product for end users and as an input for other industries. That is, a substantial share of all software produced is not developed in software companies for the general market, but rather by, or for, specific users creating custom software for their own needs.

Estimates produced early in the present decade indicated that already by that time the share of standard packaged software in overall software production was relatively small. Estimates of the share of software produced *in-house* ranged from 20% to 40% of software production and a further 40% to 50% of the market were estimated to be in *tailored software* (custom software produced for users by software service providers) (Parker and Grimm, 2000; Parker *et al.*, 2002).[39] Thus, the standard packaged software industry was only a relatively small part of the economy-wide software activity.[40]

Although software R&D expenditures outside the software sector proper are often not clearly identified, there are several national surveys that provide some insight on the relevance of software-related R&D in other industries. According to Young (1996), at the beginning of the 1990s, about 40% of services firms in Japan and Italy undertook some form of IT research activities including software development. Furthermore, about 75% of all R&D investment reported by "other services industries" was computer-related in Denmark, and over half of all R&D in the services industries was software-related in Canada. Thus, R&D and innovation in the software sector is taken up and deployed in the non-software sectors, but also firms in these other sectors have become software innovators themselves.

Today, many firms beyond the traditional software sector have embraced software development as a complement to their non-software products (Box 2.4); increased emphasis on integration of software is fuelling technological convergence between hardware, software and telecommunication technologies. Technology synergies and interdependencies across different segments are increasing and reinforcing R&D collaboration. Many non-software sector firms are engaged in some form of collaborative R&D initiatives and find that skill, experience, and cost are considered to be vital elements for the success of R&D collaboration. Beyond direct profits from any eventual product, benefits include mutual development in the area of human capital, access to intellectual property, and organisational support, among others.

Box 2.4. Embedded software development

Modern products, from aircraft and automobiles to consumer electronics, are becoming more dependent upon software embedded or bundled with hardware products. The amount of software in many products is growing at rapid pace. For example, the increased pervasiveness of digital in-vehicle components has become a main feature of today's automotive industry. It is expected that up to 100 million lines of code, double the size of Windows Vista with 50 million lines, will be used in cars by 2009 (IBM, 2004). Consequently, spending on embedded software development is increasing at a fairly rapid pace. For example, a recent survey shows that embedded software development expenditure in Japan increased to JPY 3.27 trillion in 2006, representing almost 20% growth compared to previous year, and the amount is expected to continue increasing (METI, 2007b).

Embedded software can be found in a broad range of hardware systems, supporting functions ranging from infotainment to critical systems. For many industries, software is now at the heart of new products and contributes significantly to generating revenue, increasing overall product performance and adding new functionalities. For example, in automated production processes, software contributes to increasing productivity and decreasing costs by enabling more sophisticated and integrated process control.

There are a variety of strategies for innovation with respect to embedded software. In-house development is one option that may be of particular relevance to companies with large unit volumes such as producers of mobile phones. Large volumes may permit firms to capitalise on potential economies of scale that can make such an approach cost-effective. This option may also offer companies additional strategic advantages such as the ability to hold close critical technologies that are crucial to the success of their brands or ensure quality control for technologies that are related to safety and security such as flight control systems in aircrafts and medical devices.

However, the increased complexity and diversity of embedded software applications can be demanding in terms of software expertise and other resources for R&D. As a consequence, some firms turn to collaborative approaches, such as working together to develop common platforms or, more commonly, applications, features or specific functions. This may be seen as helping to reduce overall costs and leverage strengths. One significant example of this can be found in the initiatives such as Symbian and Android that aim to develop software for mobile devices. Software companies such as Symbian and Google, mobile operating companies and many of the leaders in mobile industries, including Nokia, Sony Ericsson, Motorola, NTT DOCOMO, Samsung, LG, are making their contributions to projects to create open mobile software platforms.[41]

There are other strategies employed by firms beyond the traditional software sector in order to meet their customer demands for enhanced embedded software. In some cases, they are able to buy non-critical software components off-the-shelf or they may subcontract the development to other IT-specialised companies. At the same time, some firms that outsource embedded software development have needed to co-ordinate more closely in order to ensure delivery of convenient user interfaces, reliability, and interoperability of systems from different sources (METI, 2007).

National case studies: policy frameworks and R&D

In order to consider national frameworks for R&D in the software sector, the analysis reviewed four country case studies: India, Ireland, Israel and Spain. These are presented below. In each of these four countries, policy initiatives play a significant role in fostering the national software sector. Each country displays a mix of direct and indirect measures to stimulate R&D and investment and, in some cases, to stimulate software sector specifically.

Beyond the economy-wide framework conditions, such as the general state of the economy and the IPR regime, the case studies point to at least three other factors that have contributed to growth of the software sector in these countries:

- *Human resources*. The software sector relies heavily on human resources. The skills and expertise of workers and a relatively large pool of software specific workers, such as software developers, have contributed to the strength of the software sector in each of these countries. IT education and training are seen as very important, and language skills have further contributed to their success.
- *Internationalisation*. International linkages form part of the success of the software sector in these countries. With respect to India, Ireland and Israel the approach has been to develop a largely export-oriented software sector, with strong and competitive multinational companies, both indigenous and foreign. In the case of Spain, international collaboration has played an important role. In addition to traditional measures such as tax breaks and other financial and non-financial incentives for R&D and investment,[42] historical links[43] and migration flows have also contributed to strong international linkages. Encouragement of foreign direct investment has brought exposure to leading technologies, increasing skills levels and facilitating adaptation of best practices.
- *Clustering*. Clusters of software firms appear to be a common feature in countries where the software sector is particularly successful (like in the US, with Silicon Valley being the most notable example). In spite of the often talked-about "death of distance", proximity appears to still matter in the software sector where a flow of information and knowledge is essential. Further advantages include being able to exploit high quality infrastructures and tapping into pools of talent attracted by clusters of firms. Software firms tend to cluster in selective regions: Bangalore, Mumbai, Delhi, and Chennai in India; Tel Aviv, Haifa, and Jerusalem in Israel; Dublin in Ireland; and Madrid and Barcelona in Spain.

India

India has a very well established and renowned software sector (OECD, 2006b). Even though in some competitive dimensions India lags behind OECD economies, it has become a very successful exporter of software products, both goods and services, as well as the services of its highly skilled IT and software engineers employed all over the world. These exports do not only come from the affiliates of foreign multinationals implanted in India, but also from a growing number of indigenous firms, often multinationals themselves. Indian firms were able to gather talented workforces and deliver technical services to meet diverse customer needs. Indian firms have also been very successful in attracting offshored and outsourced IT and ICT-enabled services, such as business process services.[44]

OECD countries have been offshoring computer-related work to countries such as India and the Philippines, both of which had a good number of English-speaking graduates, as early as the 1980s. Paper documents and audiotapes were flown in for digitisation and transcription and relatively simple computer programming tasks were performed for foreign clients. Indian professionals with IT backgrounds were also brought to clients in OECD countries for software coding. However, with rapid developments in ICTs and trade and investment liberalisation, the Indian IT industry developed a "global delivery model".

In the original model, a group of Indian professionals would be sent to the overseas client and worked together with a team based in India. The professionals at the client's premises acted mainly as facilitators – they conducted negotiations, transferred information, supervised and implemented software solutions – while professionals in India provided most of the software coding and related tasks. Team members circulated between the client site and the Indian home office during the course of the project. With time, the model became more complex, with services sometimes provided from a number of locations, thus turning it into a global delivery model. While in the beginning the relative number of Indian professionals working at the client site was large compared to the number working from India, over time the onsite ratio decreased as the global delivery model matured and Indian companies became more experienced at managing increasingly sophisticated projects from overseas (OECD, 2006b).

India has also been very successful in attracting foreign multinationals wanting to benefit from local skills and cost structures. Approximately one-third of Indian exports of IT services and two-thirds of ICT-enabled services are estimated to be generated by foreign-owned companies (Business Standard, 2005). The most common type of establishment is the export-oriented affiliate. Companies that locate in India's software technology parks to serve foreign markets benefit from temporally limited but generous tax exemptions and various measures for facilitating investment and businesses' daily operations (see below).

Some foreign companies have expanded in the Indian market by buying local companies. For example, in 2004 IBM acquired Daksh eServices, India's third largest ITES (Information Technology Enabled Services) company, and in 2005 Oracle acquired i-flex, India's leading software product company. Foreign companies have also expanded in India by adopting a build-operate-transfer (BOT) model. In this case, a foreign company agrees to let an Indian IT company establish, manage and expand a unit which is taken over by the foreign company after a few years. The model is dependent on a number of contract-based criteria and tends to work best for companies producing software products (OECD, 2006b).

Government initiatives

The Indian government has been very proactive and has placed much emphasis on developing its ICT sector, and software activities in particular. However, up until the early 1980s, India focussed on building a domestic hardware industry through import substitutions, with high import duties and complex rules and regulations for importers. At that time, software was considered as a sub-section of hardware which constrained the potential growth of software industry.

This changed in 1984, with the introduction of a New Computer Policy (NCP-84), which played a significant role in developing the Indian software industry. This policy comprised *i*) a package of reduced import tariffs on software as well as hardware, *ii*) permission for foreign firms to establish themselves in India, and (*iii*) projects to set up an ICT infrastructure, including the creation of software parks.

Some of the initiatives to promote the software sector that have followed include tax and related measures, further infrastructure development, export promotion, and Open Source development. For instance, revenues from software exports received tax exemptions as part of the implementation of the NCP-84.

The Indian Government also initiated several schemes to attract and encourage (export-oriented) entrepreneurs in the software sector, including the establishment of the Software Technology Parks of India (STP).[45] STP provided office space, computer equipment and access to high-speed satellite links as well as Internet. They also provided services such as import certification, software valuation, market analysis and training. One of the key contributions to the software-exporting sector is provision of High Speed Data Communication (HSDC) services which is designed and developed by STP and is available to software exporters at competitive price (MCIT, 2007). Export-oriented firms in STP were also given a tax breaks, in particular during the first eight years of operation.

The Electronics and Computer Software Export Promotion Council (ESC) was set up to stimulate development of the software sector.[46] The ESC's primary functions are to provide member exporters with trade related information to help them participate in international trade fairs and exhibitions, and to implement government programmes. In addition, ESC has initiated special programmes for SMEs. The Council creates awareness in foreign markets to highlight the potentials of Indian SMEs, conducts market surveys and organises road shows and conferences (ECS, 2006).

The Centre for Development of Advanced Computing (C-DAC) is the core national R&D organisation of the Department of Information Technology. It is involved in the design, development and deployment of advanced information technology based solutions. C-DAC has also developed and supplied a range of high performance parallel computers, known as the PARAM series of supercomputers. The main R&D activities include High Performance Computing & Communication for scientific and business applications, Networking, Turnkey Solutions for Power, Telecom, Health, Financial Market etc., Geometrics, and eGovernance. C-DACC also runs IT-focused academic and training programmes (Advanced Computing Training School).

Special attention has also been devoted to Free Open Source Software (FOSS) with the establishment of the National Resource Centre for Free and Open Source Software (NRCFOSS)[47] in Chennai in collaboration with C-DAC and Anna University. The Centre is engaged in design and development of FOSS products and technologies with special emphasis on e-governance, school education and SMEs. NRCFOSS has developed a localised product called Bharat Operating System Solutions (BOSS) in an effort to enable India to work as a platform with a variety of languages in addition to English.

Human resources

The availability of a large number of skilled English-speaking software workers has been a major advantage to the development of the Indian software sector. Indian workers who have worked abroad and then returned to India have also contributed, especially since many acquired business networks and Western corporate culture skills that have aided them subsequently in setting up companies in India (OECD, 2004b, 2006b). Wage differentials are also often mentioned as a factor in India's attractiveness as a location for software-related activities. However, wage differentials to some extent reflect differences in productivity. Other types of costs, including overhead, organisational and transactions costs are also incurred when activities are sourced internationally and these are often relatively high in India.

Globalisation of R&D

Global sourcing of software R&D was not common until the mid-1980s when the independent software industry began to take off. Many MNEs then began to face difficulties in making in-house software development and management because hardware and software became very complex. The desktop workstation had the capacity for stand-alone programming and UNIX and the "C" programming language were widely adopted. Since then, many MNEs have started to outsource system integration and application work. Many MNEs now relocate part of their value chain to India, ranging from R&D to customer services (OECD, 2006b).

The need for global sourcing increased in the US and Europe in the 1990s, particularly as a result of Y2K preparations, ERP (Enterprise Resource Planning) installation, and the shift to online environments (*e.g.* e-commerce and e-business. The tremendous growth of this software and services segment created additional jobs for programmers, and Indian software workers had a competitive edge. Many software, and other,[48] MNEs now have a R&D presence in India (Table 2.6) and are establishing an offshore base in India for the development of application or embedded software.

Table 2.6. Some major MNC software R&D centres in India

Industry	Company	Year	Mission	Location (staff)
Software sector	Oracle (India Development Centre)	1994	Application development tools, server and platform technologies, e-business application	Bangalore, Hyderabad
	EDS (EDS India)	1996	Applications and business process outsourcing services	Chennai, Pune, Mumbai, Gurgaon (20 000)
	Microsoft (MS India Development Centre, Microsoft Research Centre India)	1998	Strategic and IP sensitive software product development	Hyderabad, Bangalore, Mumbai, Gurgaon (1 400)
	SAP (SAP Labs India)	1998	Business solution (ERP, CRM, etc)	Bangalore, Gurgaon, Chandigarh (3 000)
	ADOBE (R&D Centre)	1998	Publishing software	Noida (500)
Other sectors	Samsung (Samsung India software Centre)	2002	Embedded software for electronics(LCD TVs, MP3, etc)	Noida (300)
	Sony Ericsson Mobile Communications	2007	Mobile Phone	Chennai
	Dell (R&D Centre)	2007	Servers, storage, software	Bangalore
	IBM (Software Lab)	2001	Works on all IBM software like WebSphere, DB2, Lotus, Tivoli and Rational. The centre has added many new areas of activities such as middleware and business intelligence.	Bangalore, Pune

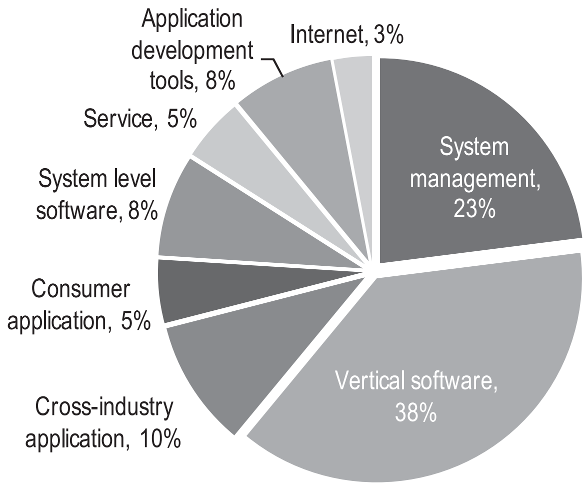
Source: OECD, based on various company reports and other sources.[49]

Israel

Israel is a relatively small economy, but it figures among the most high-tech oriented economies in the world. Israel has the highest concentration of hi-tech firms in the world second only to Silicon Valley in the United States.[50] Israel has the third largest number of NASDAQ listed companies and the largest proportion of scientists and technicians in the workforce: 140 per 10 000 workers compared with 83 and 80 in the United States and Japan, respectively.[51]

A breakdown of the software sector in Israel reveals a focus on sophisticated products and niche markets (Figure 2.12). Israeli software companies and developers focus on very complex work and produce new product market niches such as security, telecommunications, engineering, education, and anti-virus software. The Israeli software sector is also closely associated with other high-tech industries not only in the work process but also geographically with clusters in Tel Aviv, Haifa and Jerusalem. This infrastructure has contributed to the rapid development of the software sector. Israel has a long tradition in software for defence, including aviation, imaging and control applications, but the application field now extends to business sectors. Israeli application software products are found in most of high-tech industries, including biotech, telecommunication, and the Internet.

Figure 2.12. Israeli software industry export by application in 2003



Source: Israeli Association of Software House in Israel Export & International Cooperation Institute.

Government initiatives

Israel has a strong policy focus on supporting the high-tech industry, although not necessarily the software sector specifically. The *Law for the Encouragement of Industrial Research and Development* was passed in 1984 in order to promote investment in industrial research and development.[52] The Office of the Chief Scientist (OCS) in the Ministry of Industry, Trade and Labor is responsible for the implementation of government policies to support and encourage industrial R&D in Israel. The role of the OCS is to further the development of new technologies in Israel, to foster technological entrepreneurship, to support high value-added R&D to enhance the knowledge base of Israeli high-tech industries, and to promote co-operation in R&D, nationally and internationally. These policies should contribute to developing science-based, export-oriented industries to boost the economy and create employment opportunities.

The law provides a platform to expand and exploit the country's technological and scientific infrastructure and leverage its high-skilled human resources. Its most significant measures are the diverse range of programmes with financial incentives. A number of programmes have been created to specifically support entrepreneurial firms in developing high-tech innovations for commercial purposes. The OCS administers those programmes and funds R&D projects, not only for established firms but also for start-ups. There is also a specific focus on the incubation of innovation. Government support not only covers financial support, but also physical premises, tools, professional guidance and administrative assistance.

Local-level programmes include pre-seed seed, competitive R&D, and pre-competitive R&D. Some of these focus on promoting co-operation and technology transfer between business and industry (*e.g.* Magneton, Noffar). Others support technological entrepreneurship and innovation and a very early (and risky) stage of development (*e.g.* Tnufa, Technological Incubators). Some encourage start-ups (Heznek – government seed funds).

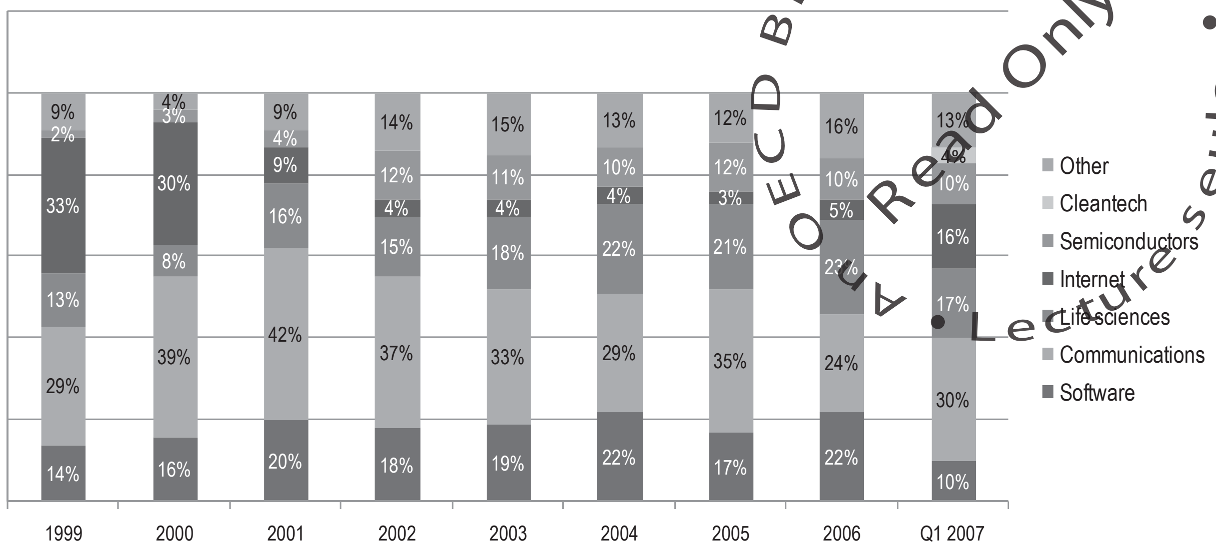
The R&D Fund, part of the competitive R&D grants, supports companies' R&D programmes. The intermediate stage between R&D and commercialisation is recognised as a very important part of the overall R&D process and is also given assistance as part of the R&D grant. If a supported R&D project leads to successful commercialisation of a product, firms have to pay royalties to the government. These are in turn re-invested to support further R&D projects. These royalties tend to be set as a percentage of the revenues derived from the commercialisation of the new product. Research Consortia and Research Institutes qualify for grants for specific R&D programmes and, in particular, the development of generic pre-competitive technologies (*e.g.* Magnet Consortium, Research Institutes).

These are complemented by a range of international (bilateral or multilateral) programmes. These include programmes that encourage the participation of Israeli firms in international R&D projects (Matimop) and ISERD, a programme that encourages joint Israeli-EU R&D ventures within the EU's Framework Programmes for R&D.

Venture capital

Venture capital is a critical element in helping start-ups bring innovative products and services to the marketplace, as illustrated by the case of the US which has a very well-developed venture capital market that contributes to US competitiveness in the software sector. Israel's well-developed venture capital market constitutes one of the main financing sources for high-tech start-ups, especially in the IT sector Figure 2.13. In the late 1970s, when the Israeli capital market was not yet well-developed, and when high risks and the relatively small scale of the market made raising capital domestically difficult, Israeli software companies had to satisfy their capital needs overseas. As a result, software firms turned to foreign venture capital markets in the US. This trend has increased over time, as the high-tech and software sectors gained in importance. Foreign capital, however, still accounts for 70% of venture funds (Baygan, 2003).

Figure 2.13. Capital raised by Israeli high-tech companies, by sector



Source: IVR Research Centre.

The ICT sector accounts for most venture capital investments. Between 1999 and 2006, the software sector accounted for an average of 18.5% of total venture capital, second to the communications sector. Innovative core technologies and market-oriented products gave Israeli software firms a competitive edge in attracting venture capital. Many Israeli software firms are now listed on the NASDAQ and Tel Aviv stock exchanges.

Human capital

The Israeli government has traditionally attached great importance to education, especially in science and technology, and to the training and maintaining of software talents. The state has also invested heavily in English as a second language, which is another vital element in the software sector's success.

The Israeli army plays a particular role in the software sector and has contributed to the creation of a new generation of high-tech entrepreneurs (NASSCOM, 2007). It provides talented young people with opportunities to acquire high levels of training in cutting edge technologies and software development. The military runs several elite units and programmes, including the software-oriented elite unit "the Central Computer Unit", or MAMRAM. The MAMRAM was created in 1960 to satisfy the specific computerisation needs in the Israeli defence sector, and since then it has grown in size and sophistication. Only those students especially talented in mathematics or science are admitted to this programme. Students receive a specialised and advanced technical training for six months and are then employed on multiple projects (Breznitz, 2002). Thus, by the age of 21, participants in MAMRAM programmes already have as much experience and knowledge as some professional career software developers. They are encouraged not only to work in areas that are closely related to military applications, but also on developing commercial applications. MAMRAM graduates form a dense network and many play leading roles in the Israeli software industry.

Ireland

Ireland is one of the world's leaders in computer and software related activities. The Irish software industry has become part of the Irish economic success story, establishing itself as one of Europe's premier locations for software development (Barry and van Welsum, 2005).

The mass-market packaged software sector in Ireland is engaged in the manufacturing, localisation and distribution (MLD) of software packages. The key players in the MLD sector (including Microsoft, Lotus, Oracle, Symantec, Informix and Corel) first established software manufacturing facilities in Ireland around the mid-1980s, duplicating and shrink-wrapping disk copies of software programs developed by the parent company and arranging for the printing and assembly of manuals. The second phase, again beginning with Lotus and Microsoft, saw these companies adding "localisation" activities – translation into other languages and cultural and technical formats appropriate to the destination markets – to their Irish operations. The third phase saw the transfer of the responsibility for distribution to the Irish operations, making Ireland an operations hub.

About half of the jobs in the foreign-owned software sector in Ireland are in highly skilled software development. This consists mostly of branches of major computing-services or IT consulting companies (including EDS, IBM, ICL and Accenture) or of activities of non-software electronics corporations, such as Motorola and Ericsson, with operations focussed on the production of embedded software and applications for products such as mobile phones. This latter segment can be seen as an unanticipated spin-off benefit from the country's success in attracting ICT hardware sectors.

The indigenous software sector produces custom software (which is provided for individual companies), niche software (written for specific business sectors) and other software services, which are provided both for organisations and consumers. The strong export orientation of indigenous firms is explained by the fact that about half of Irish indigenous software firms are engaged in the development and sale of niche products in sectors such as Banking and Finance, Telecommunications and computer/internet-based training. The emergence of this product-orientation is in part ascribable to the substantial presence of MNCs across all manufacturing and services sectors in Ireland.

The Irish software industry resembles the Israeli and Indian software industry in certain aspects. The indigenous industry's export orientation is similar to that of the Indian industry, even though the Indian software industry's export has focused more toward offshore development. Because of the limited size of domestic markets, Irish firms tend to seek foreign markets. The Irish software sector is similar to the Israeli software industry in that the focus is on providing specialised software products and support services, and avoidance of large-scale labour intensive contracting. Irish software companies have been successful in selecting appropriate worldwide niche markets, such as Banking and Finance, Telecommunications, Software tools, eLearning and computer/ Internet base training, and Online tools and applications.

Foreign direct investment and globalisation

Foreign direct investment (FDI) and globalisation, especially of services, have been a major factor in the development of the Irish software sector. Most of the world's leading software companies, including Microsoft, Oracle, and Cisco, have a base in or near Dublin. These MNEs clearly form an important part of Ireland's success in the software sector, including through their contributions to exports and employment. In addition, FDI has brought exposure to leading technologies, increasing skills levels, and adaptation of best practices. Many of those with working experiences in MNEs have subsequently become successful Irish software entrepreneurs.

The Irish government actively works to attract FDI, mostly through tax benefits, and shows strong support for the ICT sector.[53] A skilled English-speaking work force and access to the EU market further contribute to Ireland's attractiveness.

The Industrial Development Agency (IDA) promotes Ireland as a location for R&D for an increasing range of R&D functions and encourages firms in Ireland to maximise their involvement in global R&D activities to exploit international technology transfer opportunities. IDA is also responsible for providing foreign companies established in Ireland with adequate assistance, it liaises between industry and the educational sector, and it provides R&D grants. The distribution of grants is tied to well-defined objectives (*e.g.* in terms of employment, and R&D), and repayment is required in the case of an MNE's failure to comply.

In addition to the existing grants and tax incentive schemes, telecoms liberalisation led to a large drop in rates for international phone calls. This has helped create the conditions for the rapid development call centres. At a later stage, firms with call centres in Ireland added additional functions, such as financial management and software development. As a result, Ireland has established itself as an important location for "shared services" back-office activities.

Supporting institutions

Support for the software sector in Ireland includes the development of a set of supporting institutions such as the National Software Directorate (NSD), the Irish Software Association (ISA), the Irish Internet Association (IIA), and Centre for Software Engineering. These were all founded in the 1990s in response to the emergence of Irish software industry. They played an important role in the industry's evolution, notably by facilitating networking events, disseminating information, conducting strategic studies, and allocating resources such as R&D funds.

Enterprise Ireland, formed in 1997, offered diverse supports and opportunities to emerging software firms during the late 1990s. Its network includes 13 offices in Ireland and 31 around the world. It focuses on five core activities: promoting exports, investing in research and innovation, promoting the industry's competitiveness and productivity, encouraging starting up and scaling up of regional enterprise.

The Office of Science, Technology and Innovation (OSTI), in the Department of Enterprise, Trade, and Employment, is responsible for the development, promotion and co-ordination of Ireland's science, technology and innovation policy, as well as Ireland's policy in European Union and international research activities. OSTI is also responsible for basic research funding allocated to Science Foundation Ireland (SFI) and policy issues arising from Ireland's investments through SFI, and applied research and the funding for commercialisation granted by Enterprise Ireland.

The National Software Directorate (NSD) was established in 1991 and constitutes the focal point for the software industry in Ireland. The agency is in charge of co-co-ordinating the government policy towards the software industry. The primary function of NSD is to provide information and statistics related to the industry and to initiate, facilitate and co-ordinate actions which result in the overall growth of the Irish software industry.

Science Foundation Ireland (SFI) is a newly established research agency with a view to upgrade research capability into world class in niche areas of ICTs and biotechnology. SFI provides peer-reviewed grants to research centres and collaborative research between academia and industry. The funds are not given to a particular project, but rather to researchers. SFI's research funding has gradually increased from EUR 10 million in 2001 to EUR 121 million in 2005 (Enterprise Ireland, 2006). A significant portion of this budget is earmarked for Irish university-based research centres of excellence that incorporate strong involvement from MNEs and local firms.

Also, the Centre for Software Engineering was established in 1991 in the Dublin City University Campus. Its services include providing training programmes covering diverse software engineering topics, carrying out product, process and technology evaluations for companies, and organising software conferences.

National Development Plan

The government's National Development Plan (NDP) for 2000-2006 was a seven-year nationwide plan to promote Ireland's global competitiveness. This plan consisted of several operational programmes with a total budget in excess of EUR 57 billion from public, private and EU funds. The Economic and Social Infrastructure Operational Programme (ESIOP) and the Employment and Human Resources Development Operational Programme (EHRDOP) were the largest, accounting for 80% of the total expenditure.

Government funding for research was also increased significantly as a part of the NDP, even though Ireland's R&D intensity (R&D as a percentage of GDP) remained below the OECD average. Some EUR 2.5 billion was allocated to research, technology, innovation and development, a five-fold increase compared to the period 1994-1999 (Enterprise Ireland, 2006). The Irish government has now set a target to increase R&D intensity to 2.5% by 2013 (OECD, 2006c). The Higher Education Authority, through the Programme for Research in Third Level Institutions (PRTLI), constitutes the major source for upgrading Ireland's science infrastructure. The PRTL funding has enabled the establishment of 24 major research centres as well as creating some 800 post-graduate research posts (OECD, 2006c).

An independent evaluation of the NDP 2000-2006 by the Economic and Social Research Institute (ESRI) found that the NDP has been successful in enhancing the economic and social infrastructure of Ireland with major benefits to regions.

Ireland has recently announced the NDP for 2007-2013 with investment plans of EUR 180 billion (Table 2.7). There are no software-specific programmes or strategies as part of the Enterprise, Science and Innovation category. However, the software sector could benefit from sub-programmes within that category, in particular, the World Class Research (EUR 3 462 million) and Enterprise STI (EUR 1 292 million) programmes. The World Class Research Programme covers research in third level institutions and the science foundation. The Enterprise STI Programme (managed by Enterprise Ireland and the IDA) covers activities such as transforming R&D activity in enterprise, collaboration between industry and higher educational institutions, and realising the commercial potential of Ireland's research community.

Table 2.7. Ireland's National Development Plan 2007-2013 investment areas

EUR billions, current prices

Economic infrastructure Transport, energy, communication and broadband, government infrastructure, environmental services, local authority development	54.7
Enterprise, science and innovation Science, technology and innovation (STI), enterprise development, tourism development, agriculture & food development, rural social & economic development, Gaeltacht and islands development, marine and coastal communities	20.0
Human capital Training and skills development, higher education, schools modernisation & development	25.8
Social infrastructure Housing, health infrastructure, justice, sports, culture, heritage & community infrastructure	33.6
Social inclusion Children, the elderly, people with disabilities, local and community development, working age	49.6

Source: NDP 2007-2013, *www.ndp.ie.*

Spain

Computing services constitute the largest component of the domestic information technology (IT) market in Spain (Table 2.8), with sales of development and support services, IT consulting services, and operations management increasing rapidly in recent years. These sales are driven by both the business and public sectors, including through the updating of systems infrastructures and the increasing home and business penetration of Internet.

Table 2.8. Spanish domestic market for IT

EUR millions

	2003	2004	2005	2006	2007
Hardware	3 178.81	3 196.47	3 340.63	3 498.08	3 750.29
Software	1 274.82	1 360.92	1 481.50	1 600.43	1 765.75
Computing service	3 898.44	4 131.44	4 502.86	4 974.71	5 557.25
Telemetric service	587.65	668.25	770.63	848.99	938.81
Other	546.21	567.74	594.52	622.67	583.99

Source: AETIC (2007), "Las Technologías de la Información en España", Asociación de Empressas Tecnologías de la Información y Telecommuniciones de España (AETIC)/Ministry of Industry, Tourism and Commerce, Spain.

National government initiatives

Government policy to support software innovation involves several initiatives in order to achieve:

- proper broadband infrastructure for telecommunications to facilitate the appropriate use of applications and telemetric services.
- easier access for business and citizens to proper equipment (hardware) and proper applications (software).
- adequate support for companies to carry out R&D projects, both nationally and in international projects of co-operation
- improvement of the quality of software products through support for obtaining quality certifications of software processes by SMEs.

More concretely, to improve innovation performance, particularly in the private sector, the Spanish government launched the INGENIO 2010 Programme. It complements traditional science and technology policy instruments (in particular, the National R&D and Innovation Plan 2004-2007) by allocating new public funding to strategic initiatives. It aims at aligning Spain with the strategy of the European Union to reach a level of 2% of the GDP invested annually in R&D by 2010. The main strategic objectives of INGENIO 2010 are the following:

- Increase public and private expenditure in research & development & innovation (R&D&I): Reach a level of 2% of gross domestic expenditure on R&D (GERD) by 2010.
- Increase entrepreneurial participation in R&D activities: reach a 55% share of financing by the private sector in 2010.
- Work for the European Research Area, increase the participation of Spanish enterprises and researchers in the European Framework Programme.
- Finance key long-term scientific and technological initiatives, developed together by private and public entities.
- Integrate more closely universities and enterprises with the objective of 1 300 PhD holders hired each year from 2010 by the private sector through the Torres Quevedo Programme.
- Recruit and promote researchers, through the I(3) Plan (Plan for Promotion, Incorporation and Reinforcement of the Research activity), provided with EUR 130 million for the following three years to increase hiring of researchers with an accredited background.

Three new programmes are included in the INGENIO 2010 Programme:

- The CENIT (National Strategic Consortiums for Technological Research) programme launched in 2006 is intended to promote public-private partnerships and finance long-term applied research developed in co-operation between firms and public research centres. It has funded 61 projects with EUR 752 million, to be complemented with some EUR 868 million committed from the private sector.
- The Consolider programme seeks to increase the critical mass and excellence in public research centres by concentrating long-term funding on the best research teams.
- The Plan Avanza is a strategy for the development of the information society.[54] This five-year programme (2006-2010) was created by the Spanish government for development of information and communication technologies with the objective of convergence with the rest of Europe and among all regions in Spain. This plan is intended to enhance the supply and demand sides for information technology to improve the position of Spain internationally.[55] With an estimated budget of EUR 5.7 billion,[56] the plan is to increase current investment in this field from 4.8% of GDP in 2004 to 7% by 2010 (MITYC, 2007).

The Spanish government policy promotes both open source and proprietary software models. Regarding open source software, and in order to promote its knowledge and use, the Ministry of Industry, Tourism and Trade has promoted the creation of CENATIC (National Reference Centre for the Application of Information Technologies and Communication based on open sources). This is a strategic project, at national level, opened to participation by public administrations, enterprises, universities, associations of R&D&I, users and developers of this type of technology.

Among others, its objectives are to:

- Extend the opportunities that encourage open technologies in the Internet society for the benefit of society from areas such as business, research, or e-government, among others.

- Streamline open source initiatives to facilitate the development of ITC industrial infrastructure, encouraging the development of innovative projects, based on a partnership between public and private institutions.
- Facilitate the establishment of common standards in the use of open technologies to simplify their use and maximise resources.

R&D supporting programmes

In Spain, the levels of R&D and innovation activities have remained below the EU average. In 2004, R&D expenditure as a percentage of GDP was 1.1%, compared to 1.9% and 2.3% for the EU15 and OECD respectively (OECD, 2006c). There are plans however, to boost the R&D budget by 25% annually over the next four years. Spain has also created tax benefits for investment in R&D which includes corporate tax reductions of up to 40% of the Social Security cost of personnel working in R&D.

One of the main challenges to improve innovation capacity is to encourage R&D activities in the business sector, especially in SMEs which largely dominate this sector. This issue is seen as a priority by Spain's public authorities. In this context, the government approved the National Plan for Scientific Research, Development and Technological Innovation (2004-2007), which was in line with the Sixth EU Framework Programme and is partly financed by the EU Structural Funds.

Until early 2008, government programmes in support of R&D were managed by different Ministries. At the central government level, the Ministry of Education and Science and the Ministry of Industry, Tourism and Trade were responsible, depending on the area, for the management of research and technological development policies under the Programme for the Development of Technological Research (PROFIT), which is part of the National Plan. The PROFIT programme provides soft loans and grants for competitive R&D projects by firms and public research institutes. In 2007, the Ministry of Industry, Tourism and Trade provided a total of EUR 114 million in grants for the PROFT programme, and some EUR 219 million in loans were provided in 2007-2008 in ICT-related areas alone. R&D supporting mechanisms included in the National Plan have been launched to focus on several strategic sectors such as: life sciences; agro-food and environment; outer space, mathematics and physics; energy; chemistry, materials and industrial design and production; security and defence; transportation and construction; and information society technology.

There are two main national programmes specifically within the IT area: the National Programme for Computer Technology and National Programme for Information Society Service Technology. The Programme for Computer Technology is structured in nine priority areas, including software engineering and software support and development technologies. The main aim is to promote technological research aimed at the development of software technology needed for the Information Society and the development of systems and tools for the production of reliable software.

An integrated monitoring and evaluation system (*Sistema Integral de Seguimiento y Evaluacion*, SISE) has been setup for the evaluation of the National Plan, whereby groups of experts are to evaluate various elements on an annual basis and make recommendations for their continuation.

In 2007, Spain approved a new version of the National Plan R&D&I (2008-2011), with a new structure based on four main areas: knowledge and scientific and technological capabilities generation; promotion of R&D co-operation; sectoral technological development and innovation; and strategic actions. The latter includes five different strategic actions: health; biotechnology; energy and climate change; telecommunications and information society; and nanoscience and nanotechnology, new materials and new industrial processes. The R&D programme linked to the telecommunications and information society strategic action is named AVANZA I+D. In 2008, this programme devoted a total of EUR 119 million in grants and EUR 215 million in soft loans for projects to be executed during the period 2008-2010.

Spain also participates in international R&D activities, including the Eureka and Iberoeka programmes. Regarding Eureka and within the ITEA cluster, Spain is for instance involved in the COSI (Co-development using inner and Open source in Software Intensive products) project to create awareness of the industrial usage of distributed collaborative software and open source. OSIRIS (Open Source Infrastructure for Run-time Integration of Services) is another Open Source project. Many European governments and international software firms are involved in this project in order to find solutions providing high value services to customers by using the open source model. Spain funds these projects through the Profit and Avanza I+D programmes. Spain is also involved in the Joint Technology Initiatives which are being launched at the European level through joint public-private ventures (joint undertaking). In the software field, Spain is a founding member of the JU related to embedded systems (ARTEMIS) and the one for nanoelectronics projects (ENIAC).

After the March 2008 legislative elections, some new measures to foster R&D activities were approved. The most important is the creation of a new Ministry for Science and Innovation responsible for the co-ordination and management of most R&D programmes funded by the central government.

Regional situation

While the central government of Spain plays a key role in innovation policy and its implementation, regional governments have been playing an increasing role in the decision-making and management of innovation and IT programmes. Regional governments have been involved in agreements on the joint financing of the main pillars of the INGENIO 2010 plan and have progressively taken on more spending responsibilities. Many have a R&D&I plan and are investing heavily in R&D. The combination of both national and regional programmes has led to a significant increase in R&D investment in the last few years: Madrid (1.98% of regional GDP), Navarre (1.92%), the Basque Country (1.58%) and Catalonia (1.42%), are now spending more than the national average of 1.2%.

As part of this trend, science parks across the country have proliferated in recent years. Between 1997 and 2003, the number of firms and workers in technology parks tripled, while their turnover increased fivefold. In Europe, only Finland and the United Kingdom had more parks than Spain (OECD, 2006c). In the case of the Community of Valencia, for example, their emergence was stimulated by the existence of numerous SMEs working in the same sector. In the Basque region, technology centres appear to have developed to complement universities research capabilities (Inteco, 2008a, 2008b).

Some 13 000 IT related companies are currently operating in Spain, most of them (73%) in Madrid (40.8%) and the Catalonia (32.2%) regions. The trend is towards less concentration though, as there has been a significant increase in number of companies based elsewhere, especially in Valencia and the Basque region. Other communities experiencing rapid growth include Murcia, Castilla and Léon, Cantabria, Asturias and Galicia.

Software initiatives by the Extremadura regional government

The Spanish region of Extremadura is among the first regions in the world to adopt Open Source software standards in high schools and public offices (IDABC, 2003-2007). Although Extremadura had lagged behind the rest of the country in economic and technological developments, it has moved vigorously to address these gaps. Starting in the mid-1990s, the regional government decided to invest in information technology as a means to overcome its peripheral situation. As one element of the strategy, the regional government anchored the IT systems with a local version of Linux tailored to the needs of the region ("LinEx", short for Linux Extremadura).

The LinEx project is a Linux distribution created to provide all citizens with universal access to regional IT services. It focuses on specific aspects of translation and customisation. To avoid technical problems during the initial phase of the project, a Spanish company was hired to take an existing set of Linux software from the web and customise it. LinEx is specifically designed for use in regional administration and schools, but the software is distributed for free on a much larger scale than public bodies.

Notes

1. As noted in Chapter 1, OECD (2005) defines innovation as "the implementation of a new or significantly improved product (good or service), or process, a new marketing method, or a new organisational method in business practices, workplace organisation or external relations."

2. For example: Kemerer (1987); Mukhopadhyay and Kekre (1992); Banker *et al.* (1993); Maxwell *et al.* (1999); Krishnan *et al.* (2000).

3. On a theoretical level the concept of costly quality was introduced in early economic analysis (Hirshleifer, 1955). This idea was further developed in subsequent applied studies (Groocock, 1973, Freeman, 1995; Besterfield, 1979; Dale and Plunkett, 1991).

4. The OECD data presented here are based on the definition of R&D provided in the OECD *Frascati Manual* (OECD, 2002a). This manual is devoted to measuring R&D inputs. R&D covers both formal R&D in R&D units and informal or occasional R&D in other units. However, interest in R&D depends more on the new knowledge and innovations and the economic and social effects *that result* than on the activity itself. Unfortunately, while indicators of R&D output are clearly needed to complement input statistics, they are far more difficult to define and produce. In the context of innovation, R&D alone really is too narrow a measure. Ideally a broader framework on intangible investments, which covers not only R&D and related science and technology activities but also expenditures on software, training, organisation, among other areas, should be considered.

5. It should be noted that although R&D expenditure tends to be positively associated with innovation, there is not a consistent relationship in indicators for the two phenomena over time, space and fields of activity. R&D spending may vary from the actual rate of innovation, for example, due to differences in the productivity of the various R&D investments or problems in statistical estimation.

6. For the United States, statistics on funds for industrial R&D performance by for-profit businesses are provided by the National Science Foundation (NSF) through its Survey of Industrial R&D (SIRD) that covers Federal funding for that purpose, company and other funding, and total industrial R&D. The US data shown in Figure 2.2 are not for "software publishing" (NAICS 5112) also known as packaged software. Software publishing R&D funding statistics are not disclosed for the years 2002-04 and are aggregated in the Information Publishing (NAICS 511) category for 2005. The NSF data also exclude software R&D data on start-up or firms in the process of developing their first products since it classifies these early stage companies in the R&D services category (NAICS 5417).

7. For example, data are not always available at the level of the software sector, which is sometimes lumped with other activity as part of a broader category such as "computer and related activities". Furthermore, in some databases the data on the sector cover software-related activities across the economy, whereas in others it refers exclusively to the software sector.

8. R&D investment as defined for purposes of the database includes cash investment funded by the companies themselves, while excluding R&D done under contract for others or R&D done by associated companies and joint ventures. It is based on the amounts disclosed in the annual reports and accounts of the firms.

9. For the top 150 software publishers, the correlation coefficient between the aggregate R&D investment indicator and the profitability indicator is very weakly positive 0.16, though this is stronger if the calculation is made for the top 25 firms, for which the coefficient comes in at 0.53.

10. Rates of growth are shown as three-year averages since R&D flows can vary substantially from year to year as projects are completed or launched.

11. This section draws heavily on OECD (2008c), as well as other sources.

12. It is notable that in most OECD countries the availability of high-quality research personnel for the ICT industry is an increasing policy concern. See, for example, Eutema, 2007 for Austria; BMWI, 2007 for Germany; MTI, 2007 for Norway; PCAST, 2007 for the United States; MIC, 2005 for Japan.

13. This calculation includes 24 OECD member countries (Mexico, Poland, Switzerland, Spain, Hungary, Netherlands, Germany, United Kingdom, Czech Republic, Norway, Greece, Belgium, Australia, Italy, Japan, Austria, France, Portugal, Denmark, United States, Canada, Finland, Korea and Ireland) plus Slovenia. Data are partly estimated and may underestimate the actual totals.

14. This ranking is quite similar to that of 2002 (OECD, 2006a). Also, figures for non-OECD countries are available only for Chinese Taipei (41 000), Singapore (6 400), and Slovenia (500). Chinese Taipei had just slightly fewer ICT R&D researchers than France but more than Germany.

15. See van Pottelsberghe (2008) on the linkages between publicly and privately funded R&D.

16. Some private sector ICT firms do invest heavily in basic or exploratory research of their own as a means of gaining comparative advantage. For example, HP Labs recently moved to allocate one-third of its budget to such research, up from 10%. IBM is adding "big bet" projects in exploratory research and expanding collaboration, for example, with its customers (Anthes, 2008).

17. The four clusters include Cambridge and Route 128 in Massachusetts; the Silicon Hills of Austin, Texas; Champaign County in Illinois; and Silicon Valley in California.

18. Since 2000, the broader category of government budget outlays or appropriations for R&D grew by 8% annually across the OECD, from USD 197 billion in 2000 to USD 293 billion in 2006 (in current PPP USD) (OECD, 2008c).

19. GFII (2007) has produced estimates, with ongoing European Commission–sponsored projects aiming to produce further estimates.

20. All three cases enjoy high priority due to their situation at the highest policy-making level (US: Executive Office of the President, Japan: PM's Cabinet Office, EU: European Commission, adopted by the European Parliament and European Council).

21. See Chapter 1 for a discussion of intellectual property rights in the software context.

22. At the same time, patents may impact on entry and exit of firms in markets. Controlling for firm and market characteristics, Cockburn and MacGarvie (2006) find that firms are less likely to enter product classes where there are more software-related patents, although firms that actually hold such patents are more likely to enter these markets. They also find that patents also increase the chances of survival after market entry.

23. They then excluded those patents that had the words "semiconductor", "chip", "circuit" or "bus" in the title on the assumption that these patents were more likely to refer to the device used to execute the program rather than the software program itself.

24. Bessen and Hunt (2004a, 2004b) separate out IBM as they do not consider it representative of the software services industry.

25. The government of China has made it a priority to shift the economy towards higher value-added activities (Connor Linton, 2008). The recent 15-year science and technology plan calls for increase in R&D investment to 2.5% of GDP, reduction in dependence on imported technologies, improvement in the contribution to economic growth made by technological advances and joining the world's top five countries in terms of number of patents granted for domestic inventions and citations in international science papers. Software R&D figures prominently in this effort, particularly in relation to support for service industries, large-scale applications and basic software for core electronic components.

26. See Chapter 1 for further discussion of the human capital dimension of software innovation, as well as software engineering tools and the OECD business questionnaire.

27. See the note to Figure 2.8.

28. Arora *et al.* (2007), based on the Venture Economics VentureXpert database.

29. In the case of China, for example, anecdotal evidence suggests that Chinese students have good ICT skills and problem-solving capabilities. NASSCOM (2006) estimates that the suitable graduate talent pool for ICT services in China constituted over 727 000, compared to 1.7 million in India (in 2003), with "suitable" defined to mean those "with skills to be directly employed, without considering willingness or accessibility of talent".

30. Furthermore, while some of the numbers appear very favourable for some countries, *e.g.* China, when they are scaled to the population it is obvious that the potential for further growth in the diffusion of ICTs and ICT infrastructure is huge. Overall the stock of ICT-related infrastructure in countries often seen as potential recipients of offshored services activities suggests enormous potential, but there is still a long way to go before these countries, including the largest, can match OECD countries in terms of the intensity and quality of infrastructure. Furthermore, India, the major supplier of ICT-enabled services, has neither the largest stock of ICT-related infrastructure nor the cheapest broadband costs among the BRICs. The source of India's comparative advantage lies instead in the availability of ICT-trained engineers, entrepreneurial domestic firms, linguistic advantages, global ties, recent economic liberalisation, among other factors (OECD, 2006a).

31. The data for Ireland may be somewhat misleading as the majority of Ireland's software exports come from US-based MNEs who use Ireland as a base to ship products to the European Union (Arora *et al.*, 2007).

32. The computer and information services category too aggregated to provide insights into the specifics of software trade. More detailed data available for the US are available, but entail different types of measurement problems. Thus, the US Bureau of Economic Analysis (BEA) publishes unaffiliated general use computer software trade data, but aggregates the affiliated part in royalties and license fees. The BEA does not include custom software and programming services in its general use software data and does not break it out in its computer and information services data. The US Census publishes software publishing (NAICS 5112) exports data, but not imports, in its Service Annual Survey.

33. They are referenced here in order to present information on the importance of collaboration and openness in relation to other factors in the innovation process (also discussed in Chapter 1).

34. The Community Innovation Survey (CIS) is a survey conducted every four years by EU member states. The first was carried out in 1992. CIS2 took place in 1996 and CIS3 in 2001. In most countries, CIS4 was launched in 2005.

35. The closest approximation available in the CIS3 data covering 1998-2000 (European Communities, 2004) is the business services aggregate composed of computer activities, R&D, engineering activities and consultancy, and technical testing and analysis (*i.e.* NACE sectors 72+73+74.2+74.3).

36. Of which 57% were successful innovators, and 7% reported ongoing or abandoned innovation activity.

37. This is also illustrated by data from the Australian Innovation Survey where in innovating businesses with 20-99 persons and 100 or more persons, the three leading skills sought for innovative activity (in addition to general business skills) were Information Technology (25.8% and 36.3% respectively), marketing (18.5% and 26.2% respectively), and engineering (14.9% and 23.6% respectively). Data available at: *www.abs.gov.au/ausstats/abs@.nsf/mf/8158.0?OpenDocument.*

38. However, this does not provide an indication of the actual intensity of use. For an overview of studies on market penetration rates see Annex 1.

39. These figures are supported by statistics reported by the US Bureau of Economic Analysis (BEA). BEA software investment data show the following volumes for three software segments in 2006: US packaged software investment amounted USD 69.3 billion (nearly 30% of the three-segment total), in-house software USD 94.3 billion (40%) and custom software USD 71.2 billion (30%).

40. Software as a product is used and developed not just for other industries, but also for the software industry itself. This overlap is often not taken into account in the statistics.

41. More information can be found at the respective websites of these two initiatives: *www. symbianfoundation.org/ and http://code.google.com/android/.*

42. In the case of Israel, a well-developed venture capital market constitutes one of the main financing sources for high-tech start-ups, especially in the IT sector, providing an area of comparative advantage.

43. For example, colonial ties have contributed to cultural affinities between the UK and India resulting in certain educational and professional qualifications similarities, and a very good level of English for many Indian graduates.

44. India has become a major supplier of ICT-enabled services even though it has neither the largest stock of ICT-related infrastructure nor the cheapest broadband costs among emerging economies. The source of India's comparative advantage lies instead in factors such as the availability of ICT-trained engineers, entrepreneurial domestic firms, linguistic advantages, global ties, and recent economic liberalisation (OECD, 2006b). One other area where Indian firms have become leaders is in their ownership and operation of undersea cables. Collectively, Indian firms are now the largest player in this global market (OECD, 2007b).

45. Initiated in 1986, STP is a non-profit society under the Ministry of Information Technology whose primary objective is to promote and facilitate software exports from India. With over 40 centres spread across the country, STP is helping out some 6 500 software exporting companies in India.

46. ESC falls under the Department of Information Technology and has a membership base of over 23 000 exporters and manufacturers. Headquartered in Delhi, it has emerged as the centre for promotion of trade of information technology.

47. *www.nrcfoss.org.in* (last accessed 6 August 2008).

48. Such as electronics, business consulting, and financial sector firms.

49. See, for example, *www.expresscomputeronline.com/20030609/focus1.shtml* (last accessed on 22 November 2007): "MNC R&D Centres Mushroom in India".

50. See Israel Export and International Cooperation Institute, *www.export.gov.il* (last accessed on 18 October 2007).

51. According to State of Israel (2005).

52. Investment incentives are outlined in the *Law for the Encouragement of Capital Investment*, which has recently been revised to include two types of incentive programmes: grants (administered by the Israel Investment Center – IIC – which is a department of the Ministry of Industry, Trade and Labor), and an Automatic Tax Benefits Program (administered by the Israeli Tax Authorities). In order to qualify for these programs, investment projects must meet several criteria, including international competitiveness (legal definition), minimal designated investment, high value added, and registration of the company in Israel (Ministry of Industry, Trade and Labor, 2006). In order to complement this revised Law, the government has also established an Employment Grant Program which supports the establishment or expansion of industrial plants, call centres, computer service support centres and logistics centres, especially in remote areas or areas with high unemployment. Civilian expenditure on R&D in Israel is made by the business sector, general government, higher education and private non-profit institutions.

53. In 2005, Ireland's corporate tax rate was the lowest in Western Europe at 12.5%, compared to 30% in the UK, 26% in Germany, and 24% in Belgium (Enterprise Ireland, 2006). The favourable tax system has also contributed to the development of an indigenous industry as the MNEs provided both a source of revenue and access to overseas markets for domestic firms.

54. The home page for this programme can be found here: *www.planavanza.es* (in Spanish, last accessed 29 September 2008).

55. As a result of this plan, by the end of 2008, 99% of Spanish population had broadband access, 43.5% of Spanish homes and 90% of Spanish companies had broadband connections, with a connection speed up to 3 Mbps and the prices cut were by 25%.

56. During the period 2006-2007, EUR 3.9 billion was provided (EUR 2.9 billion by the Ministry of Industry, Tourism and Trade and some EUR 1 billion by autonomous communities, local entities and others).

References

AETIC (2006 and 2007), "Las Technologías De La Información en España", Asociación de Empressas Tecnologías de la Información y Telecommuniciones de España (AETIC)/Ministry of Industry, Tourism and Commerce, Spain.

Analysys International (2006), "China's software outsourcing market reached RMB 2.592 billion in Q1 2006", *http://english.analysys.com.cn/3class/detail.php?id=207&name=report&FocusAreaTitleGB=&daohang=Services&title=China's%20software%20outsourcing%20market%20reached%20RMB%202.592%20billion%20in%20Q1%202006*

Arora, A. and B. Surendrakumar (2006), "The Indian Software Industry: the Human Capital Story", Working Paper, Heinz School of Public Policy & Management, Carnegie Mellon University.

Arora, A., and R. P. Merges (2004), "Specialized Supply Firms, Property Rights and Firm Boundaries", *Industrial and Corporate Change*, Vol. 13, No. 3, pp. 451-475.

Arora, A., C. Forman and J. Yoon (2007), "Software", paper prepared for the Conference on Globalisation of Innovation: Emerging Trends in IT, Biopharma and Financial Services, The National Academies, 20 April 2007.

Arora, A., V. Arunachalam, J. Aundi and R. Fernandez (2001), "The Globalisation of Software: The Case of the Indian Software Industry", technical paper submitted to the Sloan Foundation, Carnegie Mellon University, Pittsburgh, PA.

Athreye, S. (2005), "The Indian Software Industry", in A. Arora and A. Gambardella (eds.), *From Underdogs to Tigers: The Rise and Growth of the Software Industry in Brazil, China, India, Ireland, and Israel*, pp. 7-40, Oxford: Oxford University Press.

Banker, R.J., S. Datar, C. Kemerer and D. Zweig (1993) ,"Software Complexity and Maintenance Costs," Comm. of the ACM, 36, 11.

Bardhan, A. D., and C. Kroll (2003), “The New Wave of Outsourcing”, University of California Berkeley, Fisher Centre for Real Estate and Urban Economics, Fisher Centre Research Report No. 1103.

Barry, F. and D. van Welsum (2005), “Services FDI and Offshoring into Ireland”, paper prepared for the panel session on “ICT-enabled Offshoring: Country Experience and Business Perspectives”, held as part of the June 2005 meeting of the OECD Working Party on the Information Economy, OECD, Paris; available at *www.oecd.org/sti/offshoring*.

Baygan, G. (2003), Synthesis “ Venture Capital”: Trends and Policy Recommendations”, OECD, Paris, available at *www.oecd.org/dataoecd/4/11/28881195.pdf*.

Bessen, J. and R. Hunt (2004a), “The Software Patent Experiment,” available at: www.researchoninnovation.org/softpat.pdf (last accessed 6 August 2008).

Bessen, J. and R. Hunt (2004b), “An Empirical Look at Software Patents”, Federal Reserve Bank of Philadelphia, Research Department Working Paper No. 03-17/R.

Besterfield, D.H., (1979), *Quality Control*, Prentice-Hall, New Jersey.

Breznitz, D. (2002), “The Military as a Public Space – The Role of the IDF in the Israeli Software Innovation System”, Industrial Performance Centre, MIT, April.

Business Standard (2005), “BPO Boost IT Exports to $5.2 bn”, Bangalore, 3 June, *www.business-standard.com/india/storypage.php?autono=190532*.

Charrette, R. N. (2005), “Why Software Fails,” IEEE Spectrum, September.

Cockburn, I.M. and M.J. MacGarvie (2006), “Entry, Exit, and Patenting in the Software Industry”, National Bureau of Economic Research, Working Paper No. 12563, Cambridge, Mass.

Connor Linton, K. (2008), China’s R&D Policy for the 21st Century: Government Direction of Innovation, US International Trade Commission, Working Paper Series, February, *http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1126651*.

Dale, B.G. and J.J. Plunkett, (1991), *Quality Costing*, Chapman & Hall, London.

Dempsey, B.J., D. Weiss, P. Jones and J. Greenberg (1999), “A Quantitative Profile of a Community of Open Source Linux Developers”, UNC open source research team, School of Information and Library Science, University of North Carolina at Capitol Hill, *www.ibiblio.org/osrt/develpro.html*.

EC (2006), Study on the Economic Impact of Open Source Software on Innovation and the Competitiveness of the Information and Communication Technologies (ICT) Sector in the EU, Rishab Aiyer Ghosh, European Commission, MERIT, Contract ENTR/04/112.

EC (2007a), The 2006 EU Industrial R&D Investment Scoreboard, European Commission, Joint Research Centre, Directorate General Research,Technical Report EUR 22348 EN, *http://iri.jrc.ec.europa.eu/research/scoreboard_2006.htm*.

EC (2007b), The 2007 EU Industrial R&D Investment Scoreboard, European Commission, Joint Research Centre, Directorate General Research, October, *http://iri.jrc.ec.europa.eu/research/docs/2007/sb_2007.pdf*.

EC (2007c), Analysis of the 2006 EU Industrial R&D Investment Scoreboard, DG Joint Research Centre / DG Research, Technical Report EUR 22694, European Commission.

EIU (2007), "The Means to Compete: Benchmarking IT Industry Competitiveness", Economist Intelligence Unit.

ECS, (2006), *Statistical Year Book 2005-06,* Electronics and Computer Software Export Promotion Council, India.

Enterprise Ireland 2006, Economic Profile, Enterprise Ireland 2006, Ireland.

European Communities (2004), *Innovation in Europe – Results for the EU, Iceland and Norway, 2004 Edition*, Luxembourg: Office for Official Publications of the European Communities.

Freeman, J.M., (1995), "Estimating Quality Costs", *The Journal of the Operational Research Society*, Vol. 46(6).

GFII (2007), "Étude sur la recherche et développement en sciences et technologies de l'information dans les grands pays industriels", Groupement Français de l'Industrie de l'Information, study commissioned by the French Ministry of National Education, Higher Education and Research, *www.csti.pm.gouv.fr/fr/activites-etudes.html*.

Graham, S.J.H., and D.C. Mowery (2003), "Intellectual Property Protection in the US Software Industry," in W. M. Cohen and S. A. Merrill (eds*.), Patents in the Knowledge-Based Economy*, National Research Council, National Academies Press, Washington, pp. 219-258.

Groocock, J.M., (1973), *The Cost of Quality*, Pitman Publishing, London.

Hall, B.H. and M. MacGarvie (2006), The Private Value of Software Patents, NBER Working Paper Series, Vol. 12195.

Hirshleifer, J. (1955), "The Exchange between Quantity and Quality", *The Quarterly Journal of Economics*, Vol. 69(4).

IBM (2004), "Challenges for the Automotive Industry in an On-Demand Environment", Executive Brief, *http://www-935.ibm.com/services/us/imc/pdf/g510-3956-challenges-automotive-on-demand.pdf*.

IDABC (2003-2007), FLOSS Deployment in Extremadura Spain (and various updates), posted on the website of Interoperable Delivery of European eGovernment Services to Public Administrations, Businesses and Citizens, available at: *http://ec.europa.eu/idabc.*

Inteco (2008a), Studies and Reports, León, Spain, Instituto Nacional de Tecnologías de la Comunicación, *www.inteco.es/Security/Observatory/Publications/Studies_and_Reports*/

Inteco, (2008b), "Study on Security Incidents and needs in Spanish Small and Medium-sized Enterprises", León, Spain, Instituto Nacional de Tecnologías de la Comunicación, *www.inteco.es/Security/Observatory/Publications/Studies_and_Reports/estudio_pymes_29.*

ITAA (2007), "Lower-Cost Domestic Sourcing: A Niche Opportunity for the US", Information Technology Association of America, *www.itaa.org/upload/news/docs/DX_ExecSum_Final.pdf.*

Jaumotte, F. and N. Pain (2005a), "An Overview of Public Policies to Support Innovation", OECD Economics Department Working Paper No. 456, OECD, Paris.

Jaumotte, F. and N. Pain (2005b), "From Ideas to Development: The Determinants of R&D and Patenting", OECD Economics Department Working Paper No. 457, OECD, Paris.

Jaumotte, F. and N. Pain (2005c), "From Development to Implementation: Evidence on Innovation Determinants from the Community Innovation Survey", OECD Economics Department Working Paper No 458, OECD, Paris.

Jaumotte, F. and N. Pain (2005d), "Innovation Policies and Innovation in the Business Sector", OECD Economics Department Working Paper No. 459, OECD, Paris.

Kemerer, C. (1987), "An Empirical Validation of Software Cost Estimation Models", Communications of the ACM, May 1987".

Khan, M. (2004), "Estimating the Level of Investment in Knowledge across OECD Countries" in A. Bounfour and L. Edvinsson (eds.), *Intellectual Capital for Communities: Nations, Regions, and Cities*.

Krishnan M. S., C. H. Kriebel, S. Kekre, T. Mukhopadhyay (2000), "An Empirical Analysis of Productivity and Quality in Software Products" *Management Science*, 46 (6).

Lumpkin, S., and J. Thompson (2006), The SME Financing Gap: Theory and Evidence, OECD Working Paper DAF/CMF (2006), Directorate for Financial and Enterprise Affairs, OECD, Paris.

Mann, R. J. (2004), "The Myth of the Software Patent Thicket: An Empirical Investigation of the Relationship between Intellectual Property and Innovation in Software Firms", The University of Texas, School of Law, Law and Economics Working paper No. 022.

Mann, R. J. (2005), "Do Patents Facilitate Financing in the Software Industry?", *Texas Law Review*, Vol. 83, No. 4, March 2005.

Maxwell, K., L. van Wassenhove and S. Dutta (1999), "Performance Evaluation of General and Company Specific Models in Software Development Effort Estimation", *Management Science*, 45(6).

MCIT (2007), *Ministry of Communication and Information Technology Outlook*, Indian Ministry of Communication & Information Technology.

Merges, R. P. (2006), "Patents, Entry, and Growth in the Software Industry", Working Paper, University of California, Berkeley.

METI (2007), “Outline of Survey on the Situation in Embedded Software Sector for 2007” (in Japanese), news release, http://sec.ipa.go.jp/download/200706es.php.

METI (2007a), *Interoperability Framework for Information Systems: Version 1*, Ministry of Economy, Trade and Industry, Japan, June.

METI (2007b), News release on outline of survey on the situation in embedded software sector 2007, Ministry of Economy, Trade and Industry, Japan.

MIC Japan (2005), “ICT R&D Programs for the Ubiquitous Network Society. UNS Strategic Programs”, Telecommunications Council, Japanese Ministry of Internal Affairs and Communications, *http://203.140.31.100/joho_tsusin/eng/features/r_d_progrums.pdf*.

Ministry of Industry Trade and Labor (2006), *The Israeli Economy at a Glance 2006*, Israeli Ministry of Industry Trade and Labor.

MITYC (2007), *Plan Avanza*, Ministry of Industry, Tourism and Commerce, Spain.

Mukhopadhyay, T., and S. Kekre (1992), “Software Effort Models for Early Estimation of Process Control Applications”, *IEEE Transactions on Software Engineering*, 18.

NASSCOM (2006), *Strategic Review 2006 – The IT Industry in India*, NASSCOM, New Delhi.

NASSCOM (2007), *NASSCOM – BCG Innovation Report 2007 – Unleashing the Innovative Power of Indian IT-ITES Industry*, NASSCOM New Delhi, India.

National Science Board (2008), *Science and Engineering Indicators 2008*, National Science Foundation, Washington, DC.

NRC (2003), Innovation in Information Technology, Computer Science and Telecommunications Board, National Research Council, The National Academies Press, Washington.

NVCA (2008), 2008 *Venture Capital Yearbook*, National Venture Capital Association, available at *www.nvca.org/pubs.html*.

OECD (2002a), *Frascati Manual: Proposed Standard Practice for Surveys on Research and Experimental Development*, OECD, Paris.

OECD (2002b), *OECD Information Technology Outlook 2002*, OECD, Paris.

OECD (2004a), *OECD Information Technology Outlook 2004*, OECD, Paris.

OECD (2004b), “Digital Broadband Content: Mobile Content, New Content for New Platforms”, OECD, Paris.

OECD (2005), *Oslo Manual: Proposed Guidelines for Collecting and Interpreting Technological Innovation Data, 2nd edition*, OECD, Paris.

OECD (2006a), *OECD Information Technology Outlook 2006,* OECD, Paris.

OECD (2006b), *Innovation and Knowledge-Intensive Service Activities*, OECD, Paris.

OECD (2006c) “Policy Mix for Innovation in Spain: Key Issues”, OECD, Paris.

OECD(2007a), *Open Innovation in Global Networks*, OECD, Paris.

OECD (2007b) “Global Opportunities for Internet Access Development”, COM/DSTI/DCD(2007)3, OECD, Paris.

OECD (2008a), “Innovation in the Software Sector: Business Questionnaire” and associated responses.

OECD (2008b), “Intellectual Assets and Value Creation: Synthesis Report”, OECD, Paris, available at *www.oecd.org/sti/ipr/iavc*.

OECD (2008c), *OECD Science, Technology and Industry Outlook 2008*, OECD, Paris, *www.oecd.org/sti/outlook*.

Parker, R. and B. Grimm (2000), “Recognition of Business and Government Expenditures for Software as Investment: Methodology and Quantitative Impacts, 1959-98”, US Bureau of Economic Analysis Working Paper, available at *www.bea.gov*.

Parker, R., B. Grimm and D. Wasshausen (2002), “Information Processing Equipment and Software in the National Accounts”, Bureau of Economic Analysis Working Paper 2002-02, Washington DC, available at *www.bea.gov*.

van Pottelsberghe, B. (2008), “Europe’s R&D: Missing the Wrong Targets?”, Bruegel Policy Brief, Issue 3.

Price Waterhouse Coopers (2007*), Money Tree Report, www.pwcmoneytree.com/MTPublic/ns/index.jsp*.

Smith, B.L., and S. O. Mann (2004), “Innovation and Intellectual Property Protection in the Software Industry: An Emerging Role for Patents”, *University of Chicago Law Review*, Vol. 71, pp. 241-264.

State of Israel (2005), “Invest in Israel –Where Breakthrough Happens”.

Teece, D.J., G. Pisano and A. Shuen (1997), “Dynamic Capabilities and Strategic Fit”, *Strategic Management Journal*, Vol. 18.

Thoma, G. (2008), “Some Stylized Facts on the Spatial Distribution of Innovative Activities: New Evidence from Patent Data”, OECD, Paris.

Van Welsum, D, and T. Xu (2007), “Is China the New Centre for the Offshoring of IT and ICT-enabled Services?, DSTI/ICCP/IE(2006)10/FINAL, Directorate for Science, Technology and Industry, OECD, Paris, available at *www.oecd.org/sti/offshoring*.

Von Hippel, E. (2005), *Democratizing Innovation*, The MIT Press, Cambridge, MA.

Young, A. (1996), “Measuring R&D in Services”, STI Working Paper, OCDE/GD(96)132, Directorate for Science, Technology and Industry, OECD, Paris.

Chapter 3

User Perspectives on Software Functionalities

As the importance of software increases and has become social and economic "infrastructure," factors such as security, privacy, mobility, reliability, accessibility, and interoperability are becoming increasingly important aspects of software. This chapter explores how these factors are considered in the innovations and development process.

Introduction and main findings

As software becomes more deeply embedded in social and economic infrastructure, the range of functionalities sought by users is broadening and the exigencies for performance with respect to the various functionalities are increasing. In some cases, software innovators are creating new markets (as happened in recent years with on-line music sales), while in other cases the demand is user-led with software developers responding with new or improved products (as happened with security enhancements in home computing systems). New or enhanced functionalities are sometimes introduced by the original software developers, but also by follow-on developers writing applications or even modifications of the original product.[1]

Both original and follow-on developers may bring new functionalities for existing products to the market, incrementally or individually. In some cases, developers and platform owners also move to bundle in expanded sets of functionalities – including via acquisition of those developed by add on innovators. These may be made available via periodic major new releases of the software products. This bundling in of existing functionalities, in turn, may spur new rounds of innovation in functionality by developers aiming to generate renewed traction in markets.[2]

The various functionalities of a software package together shape the user decision to employ the product; they also shape the user experience. Generally, multiple functionalities are sought from individual products. In many cases, the specific functionalities are not an end but rather a means to obtain a desired aspect of performance from a software product. For example, a user may seek reliability in a software package, but ultimately have the objective of using the software for a more concrete purpose. For example, a software package designed to monitor a critical piece of safety equipment may also need to meet very high standards in terms of reliability.

This chapter addresses user perspectives and functionalities by considering an illustrative set of functionalities. This set highlights examples that have become priorities for many users as well as the manner in which the software sector has innovated to deliver them. These include security and privacy, mobility, interoperability, accessibility and reliability. Annex B considers how software is transforming industries beyond the traditional computer industry, in part by adding new functionalities to software in their products.

Main points from the analysis include:

- As economy-wide activities are deeply affected by software, issues relating to software functionalities have come to concern stakeholders across society. Software functionality can have systemic effects with respect to the economy (*e.g.* enabling new forms of organisation) and society more broadly (*e.g.* social inclusion).
- Market demand for software functionality plays an important role in propelling technological innovation by providing signals and incentives for innovators to act. At the same time, the nature of software as a digital, non-rivalrous product means that there can be large returns to scale for innovators that are able to respond to this demand.
- Due to the technologically heterogeneous and complex nature of software functionality, there is an increasing emphasis on collaborative and, in some cases, open innovation approaches to development of improved functionality. The diversity in the content and in the technologies means that individual firms or developers face challenges in delivering comprehensive solutions and generally must draw on resources beyond the walls of the firm in order to assemble the necessary elements for success.
- The contribution of user knowledge and experience to the development of software functionality has emerged as an important source of input for innovation processes; the software sector is a leading sector in the engagement of users, a factor that is contributing to the dynamism of the innovation activity.
- The amount of software in many modern products, from automotive to consumer electronics, is growing at rapid pace, and for many industries software has become the heart of their new products. Software has become a differentiator and enabler overall increases in product performance and new functionality. Moreover, software developers are reaping the benefit of large economies of scale and low marginal costs by marketing their products, adapted, to many other sectors including reuse of embedded software.

Software functionalities and user perspectives

This chapter considers five selected functionalities in order to illustrate the types of innovation underway in software, taking into account user perspectives. The functionalities covered include security and privacy, mobility, interoperability, accessibility and reliability.[3] The chapter reviews the definition of these functionalities and examines issues surrounding their evolution including user participation (Box 3.7). Annex B underscores the role of businesses beyond the software sector as users and developers of software functionalities.

Security and privacy[4]

Background

Over the last 20 years, information and communications technologies (ICTs) have become essential for governments, businesses and individuals in most economic and social activities. Information technologies and the use of the Internet in particular provide a powerful driver for innovation, growth and social well-being. As a result, economies and societies increasingly depend on ICTs. For example, which key activity, public or private, could be carried out today if its information technology component were not available? Which business sector could continue to operate, which government branch could continue to provide services if suddenly the Internet were to collapse? Even critical infrastructures (CI) – *e.g.* water, energy, transport – rely increasingly on the effective functioning of ICTs. Information security and privacy protection are two important and sometimes inter-related policy areas in relation to ICTs.

With regards to information security, our societies' reliance on ICTs and the inter-dependence of users due to the generalisation of Internet usage was recognised by OECD member countries in the *Guidelines for the Security of Information Systems and Networks* ("Security Guidelines", OECD, 2002). These guidelines aim to promote a culture of security among all participants as a means for protecting information systems and networks, from government infrastructures to global corporate systems and home personal computers. The guidelines provide a set of high level policy and operational principles that create a general frame of reference to help participants (whether government, business and individual user): *i)* understand security issues; *ii)* respect ethical values in the development and implementation of coherent policies, practices, measures and procedures for the security of information systems and networks; and *iii)* take responsibility according to their role. Since the adoption of the guidelines, governments and businesses have developed policies and initiatives to foster a culture of security for information systems and networks and have implemented a number of measures to support them (OECD, 2004a, 2005; BIAC/ICC 2003, 2004).

Ensuring ethical use of ICTs and respect for fundamental values is another essential objective. When organisations use software to process data relating to identified or identifiable individuals (personal data), individual's privacy and liberty can be impacted. The OECD *Guidelines on the Protection of Privacy* ("Privacy Guidelines", OECD, 1980) represent a long-lasting consensus on a set of fundamental principles to enable the economic and social potential of information technologies to be realised while protecting privacy. These Guidelines are reflected in numerous national and regional legislative and regulatory instruments, as well as self-regulatory privacy codes of conducts, practices and policies, in both OECD and non-OECD countries. Effective implementation of these privacy frameworks is however challenged as ever more information systems are connected to the Internet, data processing routinely takes place anywhere in the world, personal data flows across borders, is stored, mirrored and processed in more distributed ways than ever before.

Both the OECD Security and the Privacy Guidelines reinforce each other: the Privacy Guidelines include a security safeguards principle that the Security Guidelines contribute to addressing.[5] The Security Guidelines include principles that aim to ensure that security measures support and remain compatible with essential values such as privacy.[6] Both are considered essential by OECD countries, as evidenced by the draft Ministerial declaration for the Seoul Ministerial Meeting on the Future of the Internet Economy (17-18 June

2008) and its supporting report, "Shaping Policies for the Future of the Internet Economy" (OECD, 2008a).

Software security and privacy

Software is one of the main components of ICTs. Software determines what the system does and how it is done, which is essential with regards to both security and privacy protection: whether and how data, including personal data, can be collected and stored, how it is processed, whether it can be linked to other data, shared and with who, how long it will remain in the system. All these aspects and many others are determined by software functionalities. Software privacy may not be sufficient to ensure full privacy protection, but it can play a significant role to provide more efficient privacy protection.

Although often tied together, security and privacy are two separate and independent functionalities. Depending on the needs of the user security and privacy may be related, but it is far more common that they are separate, yet vitally important functional requirements relied upon by users of many different kinds of systems. As discussed more fully below, each user faces choices about what functions are needed, and what risks can be assumed, and privacy and security, like the other functional requirements which follow in this paper, are important, yet distinct, functions.

Software security is also essential as a large number of threats seek to exploit software vulnerabilities or vulnerabilities resulting from software deployment and configuration. For example, one of the main threats to the Internet Economy and to critical information infrastructures is the development of malicious code called Malware, including viruses, worms and trojans. Malware has evolved from occasional "exploits" in the mid-1990s to a global multi-million dollar criminal industry. It is a form of software that is inserted in information systems exploiting software vulnerabilities and that could be used to launch cyber attacks for money extortion, information theft (*e.g.* identity theft), espionage or – hypothetically – terrorism (OECD, 2008b).

"Privacy" and "security" in the context of software can be interpreted as meaning software with embedded privacy and security features (*e.g.* access control features like password protection), as features that are coded in a secure or privacy-friendly way (*e.g.* via secure coding or privacy by design approaches) or as specialised functional software (*e.g.* a firewall or antivirus software package).

Risk and risk tolerance

Risk can be defined as a level of probability that negative consequences or impact on an activity could occur. Any activity carries with it a certain level of risk. Security and privacy are a means to manage some of this risk with the objective of maintaining each discipline at an acceptable level.[7] They are implemented by adopting measures (also called "controls") to prevent and mitigate risk. Nevertheless, whatever the measures taken, risk can never be completely eliminated. Absolute security and privacy do not exist. For any activity, there is always a remaining level of risk that must be accepted. Otherwise, as is often quoted, "the only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards – and even then I have my doubts" (Eugene H. Spafford).[8]

An important driver of demand for security or privacy in software is the acquisition of a greater degree of control over perceived and real risk related to the use of the software. There is always a requirement to balance business, personal, or governmental needs, including costs and other factors versus safety, security, privacy, and other functionalities (*e.g.* performance, accessibility, connectivity, scalability, interoperability) and the remaining risks must be acceptable to the user. However, there is a security paradox[9]: the user may become less tolerant of remaining risks, exactly because things have become more secure. The more preventative measures are taken, the less tolerance there may be for remaining risks and unforeseen disasters. Yet, beyond a certain point, costs may become prohibitively expensive relative to the benefits in terms of increased security.

Risk may be seen as the probability of a given event associated with exposure to negative impacts, in this context resulting from exploitation of vulnerabilities or weaknesses. When both the probability and the impact are low, users are usually unconcerned, feel secure and accept the risk. When both the probability and the impact are high enough, users become concerned and can be motivated to take measures to reduce or control the risk.

In business terms, the level of acceptable risk is known as the "risk tolerance". The overall risk tolerance varies according to an infinite list of parameters ranging from the specific context, purposes and expected benefits from the use of the software to intangible and unquantifiable factors such as societal, cultural and individual values and beliefs that vary across the world. The tolerance for tangible risk elements such as online fraud can be quantified as direct measurements can be made, but it is not the case for many other components of a given user's risk tolerance, such as the disclosure of personal information. At a macro level, only very general trends can be identified.

Information security and privacy

Information security traditionally relates to the confidentiality, integrity and availability of information. In software development, confidentiality is most commonly applied to access controls, the protection of stored data and data communications; integrity is most commonly applied to software processes and the data being processed; availability is often expressed in terms of the software processes and information being available when and where required, often blending both software processes and telecommunications delivery mechanisms. In some instances, availability is related to interoperability between programs sharing and exchanging data. Closely linked to these three dimensions are audit and appraisal processes that can lead to some level of assurance. The level of assurance is a measure of the comfort the security and privacy controls provide. Audit can give some comfort that the controls are working properly. Appraisal can give comfort that the security is adequate. The OECD *Guidelines for the Security of Information Systems and Networks: Towards a Culture of Security* (OECD, 2002) provide a set of nine principles that apply to all participants (*i.e.* businesses, governments, individual users) at all levels to promote a culture of security as a means of protecting information systems and networks.[10]

Privacy has several interpretations based on cultural and legal understandings. In its simplest form, it can be expressed as an internationally recognised fundamental value: respect for and protection of privacy, including the protection of personal data. Privacy can also be expressed as the ability of an individual or groups to seclude themselves or information about themselves and thereby reveal themselves selectively (privacy in this case is often referred to as "the right to be left alone"). In software terms, privacy protection is most often linked to the control of personal data. Personal data is defined as information relating to identified or identifiable individuals. Control includes the limitation of collection, use, storage, disclosure to third party, linkage with other data, etc. The OECD *Guidelines on the Protection of Privacy and Transborder Flows of Personal Data* (OECD, 1980) set out core principles to assist government, business and consumer representatives in their efforts to protect personal data. They represent an international consensus on general guidance concerning the collection and management of personal information.

Ultimately, from a software innovator's perspective, software security is about identifying, implementing and operating the technical measures that will control and mitigate risk to reduce it to an acceptable level to the user. While, from the same perspective, the same logic can be applied to privacy protection, in reality, software privacy is often about translating into technology high level principles such as the one set forth in the OECD Privacy Guidelines or in national laws. Technology choices within a company usually follow on from the corporate operating security and privacy policies that, in turn, are derived from the high level business principles followed by the organisation.

From a business perspective, security can be more easily connected to the overall objectives of the user or his organisation than the benefits of investments related to privacy protection, as it is much more difficult to rationalise from the sole perspective of the interest of the data controller.[11] This difference may explain why the frameworks for privacy protection are often provided by law or government policy.[12] However, the benefits of security spending are sometimes difficult to measure because they relate to uncertainties (risks). In the instance of privacy, a business or government deciding what needs or functionalities are required faces somewhat different type of problem. For example while a business may not protect the integrity of its customers computer systems, it may have a requirement to protect the privacy of its customers.

Security and privacy controls

To reduce the level of risk, there are three general types of security controls and privacy controls: technical, administrative and physical. Their objectives may be to prevent, detect or correct risks. Technical controls correspond to technology measures. In a software context, they include the security or privacy features of the software itself, *e.g.* password access control features or audit log capacity. Administrative controls concern the manner in which the system is managed through policies, procedures and standards in relation to the operation of the system (*e.g.* network administrators setting policy for machine interaction, or password management policies, backup recovery procedures or analysis of audit logs). They also correspond to actions performed by users of the software (*e.g.* users choosing and remembering strong passwords and keeping them secret), or system administrators analysing audit logs generated by the software. Physical controls rely on the application of risk reducing physical barriers and deterrents such as alternate power sources or data backup devices. A holistic approach that uses all three types of controls has to be applied to reduce risk as security is only as strong as its weakest part.

Users commonly perceive a software as secure or privacy protective when it includes a number of security or privacy features. However, from a more comprehensive security and privacy perspective, the level of risk is appropriately reduced only when the security or privacy features included in the software are *operated* within a broader security *management* strategy that reflects the acceptable level of risk for the user. This implies that the user:

- has an accurate understanding of the risk in terms of threats and vulnerabilities, and their potential negative consequences and impact on their activities,
- knows the level of risk that is considered as acceptable,
- applies the technical measures that can prevent and mitigate risk to reduce it to that level, and
- implements the operational and the management measures that can provide the appropriate context for the technical measures to be efficient.

Software security and privacy are sometimes seen to relate primarily to the technical dimension of security and privacy. This is an important dimension, but the simple existence of technical software security controls does not as such reduce the level of risk effectively unless these technical measures are operated and managed according to good practices. For example, a given software application may include a password feature but if users choose weak passwords such as their name or birthday, or if they keep their long passwords handy on a sticker under their keyboard, it is unlikely that risk be reduced to an acceptable level. Similarly, audit log features that keep track of all the actions performed by the software and users of the software can be embedded in the software, however the audit logs generated have to be analysed by a skilled administrator to detect unusual patterns, errors or possible attacks.

In general, ensuring adequate software security entails costs for system operators in all three control elements: technical, administrative and physical controls. Investment in education of users may be costly, but also has the potential to yield some significant reduction in these expenses.

Functional requirements

A key element for users is the assurance that the software code is fit for the purpose for which it was intended. At a high level, the goal to protect security or privacy seems simple, though, in practical terms, it requires a precise understanding of the risk context, or risk assessment, which often involves resources and planning. Determining how "fit" is measured is even more difficult because it has to "fit" the business scenario and perceptions of the risk tolerance of the user. Users will balance a wide range of critical or important functional requirements, such as mobility, scalability, interoperability, security, privacy, or ease of audit, just to name a few. In specific circumstances, such as banking, these can be better defined and software developed and tested to fit requirements. In general scenarios, the looser the "fit" then the less effective software becomes and the more difficult to link it to operational environment. On the other side, there is a danger that if security and privacy are too tight, the software will become expensive or restrictive.

Audit processes ensure that the controls in place are operating as they are intended and appraisal processes ensure the controls are fit for the purpose intended. Where software is used in a defined environment, such as in a business, audit and appraisal can be considered as a possible administrative option. But in the consumer market where the consumer tends to invest only the minimum in security or privacy management and operation, they are not practical. That said, much has been done to ensure that these functions are made more available and in a more useable manner for general users. Internet service providers (ISPs), for example, have sought to embed more security protections into their underlying service offerings, which allows the ISP to provide the audit and appraisal functions on behalf of its customers.

Role of security and privacy features in users' software choice

Security or privacy software features certainly affect the user's choice, but to an extent that is hard to measure. It is difficult to distinguish between the user being motivated to select one piece of software rather than another for the embedded security features or privacy features as opposed to other features. The primary reason for acquiring software is usually not its security features or privacy features but its capacity to perform tasks that meet user's needs. The extent to which users are ready to dedicate money and resources for security or privacy to perform these key tasks depends on a large number of variables, ranging from level of comfort with technology to purpose of purchase. Security or privacy are only subsets of the user's overall software cost and benefit of ownership, subsets that can include both costs for the acquisition of software and for its management, operation and support. Furthermore, as noted above, security and privacy management costs are usually much higher than software acquisition cost for the user. Ultimately, there is no simple methodology to measure, within the overall software cost of ownership, how much users consider the investment for software security or privacy – among other features – to be worth.

Trends

Several major trends related to the evolution of ICTs and ICT usage can be identified that might influence the role played by security or privacy in software:

- ICTs have become ubiquitous and critical in most people's lives.
- ICT connectivity and mobility are going global.
- Software is shifting from a product to a service, in some areas.
- Crime has migrated on-line.
- Software is growing increasingly complex.
- Users are exhibiting reduced tolerance for risk.

Supply of security and privacy functionalities

On the supply side, a number of processes and practices exist for developers and innovators to enhance security or privacy features in software, such as code evaluation practices, design methodologies, better coding practices as well as standards and other security or privacy assurance mechanisms. Many of these processes and practices are actually increasingly used by software developers (*e.g.* code evaluation, better coding practices), others are limited to niches where security assurance plays a particular role (*e.g.* security for critical infrastructures in the defence sector). The factors enabling or impeding their use are varied and complex.

Risk is a function of threats exploiting vulnerabilities and, while there can be vulnerabilities at all levels including human (*e.g.* poor passwords), procedural (*e.g.* poor training) and organisational (*e.g.* poor oversight), a major source of vulnerability is related to the quality of the software code. While it could be extreme to consider that security or privacy in software is "all about coding"[13], it is recognised that good coding practices and code evaluation are essential elements for both security and privacy. What is important, however, is that an efficient process for code evaluation and vulnerabilities identification and resolution is in place.

Government policies can be an important driver to protect the security of information systems and networks and the privacy of individuals although it is unclear how much they foster software innovation for security and privacy. Procurement processes for government and private sector acquisition can also play a role in improving both privacy and security. A better understanding of the underlying incentive structure for security and privacy in software could certainly help foster software innovation in this area. Tailored policies to support innovation in the area of software security and privacy could create positive externalities and help develop a culture of security and privacy and, more broadly, foster trust and confidence in the Internet economy.

User expectations

Understanding individual privacy and security needs is challenging. Large organisations can affect resources for this exercise, formulate their requirements in relatively formal terms and assess the efficiency of the measures in place through audit and appraisal. It may be that smaller organisations and home users sometimes fall back on the basic expectation to have the same level or better security or privacy in a digital environment as enjoyed in their physical one. In business terms, this equates to having a similar risk appetite in both environments. They may have a tendency to expect technology to deliver what they need without them having to define it and they often have a heightened awareness and sensitivity to threats and vulnerabilities without knowing what to do.

Software innovation takes place at the functional and operational levels on the supply side of markets as a consequence of user demand. Fortunately, companies often place many savvy technical and consumer-oriented employees in key roles to ensure that end users do not have to express their concerns or needs directly to developers, or *vice versa*. Companies which focus on cybersecurity and usability for end-users will benefit from meeting those marketplace needs.

As the software paradigm shift takes hold and some forms of software increasingly move from being a product to being a service, all participants will need continuous education in order to safely exploit the opportunities emerging, including transferring individual skills on personal risk management in the physical environment to the digital one. These aspects of security and privacy in software are about changing individual behaviours and creating cultures of security and privacy, as well as technological responses by software developers to needs articulated through market demand. Creating cultures of security and privacy that change behaviours regarding the use of information technologies is also about users adopting a habit of “learning for life”.

Reducing this gap requires raising awareness and education of all participants. Individual behaviour might also be motivated to change by appropriate policies and technology measures (*e.g.* security and privacy usability). Changing individual behaviour to adopt a culture of security and privacy is, in any case, a long term challenge.

Main elements and summing up

Software can include a wide range safeguards and controls to protect privacy as well as prevent and mitigate security risks. Security and privacy yet somehow different functional requirements are important dimensions for software users, playing an essential role for economies and societies that are increasingly reliant on ICTs, including for their critical infrastructures, day to day business, government operations and individual activities. Software innovation in these areas can directly benefit software market players at a micro level and can also indirectly improve the overall level of trust and confidence in the Internet economy and society and the economy as a whole.

- Users' demand for both security and privacy in software depends on the level of risk that they are ready to tolerate ("risk tolerance") in each area. Users' risk tolerance varies according to a large number of context-specific factors, including societal, cultural and individual values.
- Security features and privacy features embedded in software play important role for risk reduction but they need to be operated and managed according to good practices and within an overall risk management strategy.
- Security and privacy software features affect users' choice but to an extent that is hard to measure. Often, privacy or security are not the primary reasons for selecting a software package and the relevant features may not be easily distinguished from other secondary software features in some products. With disparate investments into processes (*e.g.* audit) and planning leading to a precise understanding of their risk context, in some instances it can be difficult for users to assess properly whether the embedded security and privacy features in software are "fit for purpose", leaving them exposed to unacceptable privacy or security risks.
- A few general trends seem to indicate a likely decrease of users' tolerance to risk related to software: *i)* the increased dependence of all users on ICTs for their essential activities, *ii)* the shift from software as a product to "software as a service"[14], *iii)* the migration of crime online (*e.g.* fraud via malware), *iv)* increased user awareness for security and privacy following government policies such as data breach notification legislation.
- Quantifying security and privacy in relation to software is challenging: privacy and security metrics are tied to many difficult to quantify or variable parameters.

Elements on processes for software security and privacy

- Many different layers of an organisation are involved in the development and integration of both security and privacy capabilities into software and involve different teams working together to ensure that the user can obtain the best experience from software : *i)* the overall objectives of the organisation are expressed at the business level, *ii)* the operational level defines why the functionality is required, *iii)* the functional level defines what functionality should be provided and *iv)* the software development level determines how the functionality is provided. In some cases, software innovation takes place at that bottom level, whereas other business models integrate security and privacy more thoughtfully into the core objectives of the business, which translates into improved development, operation and functionality.

- Secure and privacy-friendly software can be developed using waterfall or iterative software development models, each having their strengths and weaknesses. "Security and privacy by design" approaches can also be adopted, as encouraged by the OECD 2002 Security Guidelines and recent OECD work on privacy. In the area of privacy, there are some mechanisms to help software developers check their products against privacy protection criteria to assess the effectiveness of their design and protect privacy from the outset.[15] However, so far these standards are not globally adopted and utilised. Furthermore, though there is a certain degree of consistency at a high conceptual level, privacy related regulations vary across countries creating challenges to the implementation of privacy requirements at the practical software engineering level (OECD, 2002).
- Code development and evaluation processes play a key role to eliminate vulnerabilities and should not be confused with the software licensing regime.
- Both privacy and security assurance mechanisms might be enhanced by following international guidance, good and best practices, as well as formal and informal standards. Self declarations, certificates, trustmarks and seals might also be used to achieve this objective. While privacy requirements are often expressed in general, legal and technology neutral terms, there seem to be no definitive software/ technical processes, methodologies or mechanisms to translate these concepts into functional technical requirements that can be implemented by software developers. However, there are now privacy seals available for software, available though some initiatives recently launched in this area. In the area of information security, a number of security assurance mechanisms exist to provide assurance regarding the integrity of the software (*e.g.* code signing) or assurance that the security can be demonstrated to be fit for purpose, through international and national standards, reputation of the vendor, vendor and third party certification schemes.

Mobility

Introduction

Electronic delivery of information and access to digital content are becoming ubiquitous, driven by the technological advancement of broadband wireless networks, platforms and software. Network convergence and widespread diffusion of high-speed broadband have created new opportunities for innovation in software. Platforms are increasingly mobile with respect to points of access to the fixed infrastructure and are evolving with some independence from specific systems. The emphasis on mobility is reshaping the ties between the software components and the network hosts where they execute, resulting in software that retains a higher degree of bandwidth utilisation, flexibility and robustness than in the past.

These developments have opened up space for new market demand, business opportunities, productivity and innovation. Across the economy, users are demanding computing structure that is mobile and available at any time and any location.

Types of software mobility

Mobility is an important quality for products in the current environment where information processing has been thoroughly integrated into everyday objects and activities. Experience over the last few years with mobile data and information communications has led to increased demand for mobile applications and services for the general public as well as business, especially in those business processes that involve workers on the move. Developments in middleware are supporting the improved integration of wide ranges of mobile networks and computing devices.

There are different kinds of mobility schemes, such as terminal mobility, personal mobility, and service mobility. From the software perspective, mobility in a ubiquitous system includes physical mobility and connectability of devices and non-physical modes such as the ability to function in diverse environments.[16]

Main drivers for mobility

Enhanced broadband and wireless networks

One of the key factors enabling mobility is the increased capability of broadband wireless networks. Until recently, mobile networks did not have enough capacity or sufficient bandwidth to guarantee a good user experience with respect to sophisticated digital content. Now, with heavy investment in broadband infrastructures, especially among OECD member economies, high-speed digital networks are becoming a reality. In particular, implementation of 3G mobile telephony is well underway and the number of subscribers to 3G services is expanding (Table 3.1).[17]

Table 3.1. Current 3G subscribers

System	Number of countries	Number of subscribers
cdma2000	72 (mainly America, Asia)	275.2 million
W-CDMA	55 (mainly West Europe and Japan)	70 million
HSDPA (3.5G)	36	N.A.

Source: OECD (2006), *Innovation and Knowledge-Intensive Service Activities*, OECD, Paris.

The current 3G deployment provides greater bandwidth and increased functionality for delivery of content in a mobile fashion. Already as of the first half of the present decade, 3G networks were delivering access speeds ranging from 128 Kbps to nearly 2 Mbps (OECD, 2004b). With speeds greater than 200 kbps, users are able to rapidly perform such tasks as downloading music albums, conducting video conferences and playing interactive games. Enhanced broadband capabilities facilitate convergence of data, video, Internet and multimedia services, and these services are now available over a broader geographical area.

Other wireless technologies are also being developed at rapid pace, facilitating connection of a broad range of electronic devices, including over wide and local areas. Examples of particular relevance in this context include WiFi (wireless fidelity), WiMAX (Worldwide Interoperability for Microwave Access), iBurst and WiBro. For example, WiMax can deliver last-mile wireless broadband access without the need for direct line-of-sight to a base station. Although it is in the beginning stages of commercialisation, some services are already in place. In Korea during 2006, telecom companies launched "WiBro" services around the Seoul area, offering connectivity even when the user is moving, but the take up has been slow.[18]

Technical advancement of mobile platforms

Mobile device manufacturers are working with content providers, software industries and other industry participants to develop handsets and features that facilitate access to and use of mobile contents. Mobile devices such as smartphones, PDAs (personal digital assistants), GPS electronics (global positioning system) and handheld games have gained computing power and functionality. For example, mobile phones today provide multiple arrays of functions such as voice, data, music, photographs and games. Smartphones, devices that while providing voice telephony, also run an operating system allowing developers to code to the machine level and control every facet of the device, are proliferating. A typical smartphone or PDA has a processing capability equivalent to personal computer back in 1990s. IDC, a provider of ICT market intelligence, forecasts that shipments of converged mobile devices (*e.g.* smartphones and PDAs) will grow from 124.6 million in 2007 to 376.2 million in 2012.[19]

Several different operating systems on mobile devices have been developed with a variety of client-side execution environments such as Windows CE, Symbian and Palm OS. Shipments of PDA devices loaded with mobility-friendly operating systems are tending to rise (Table 3.2). In the case of mobile game devices, programs such as Java 2 Micro Edition (J2ME) and BREW have been developed in order to optimise for use in small devices. Because of the small screens in mobile devices, software firms are developing special software (*e.g.* Opera Mobile, Scope, jig browser) that allows users to look at the Internet more easily and to provide full browsing function.

Table 3.2. Worldwide preliminary PDA vendor shipment estimates by operating systems, 1Q2007

	1Q07 (shipment, units)	1Q07 market share (%)	1Q06 (shipment, units)	1Q06 market share (%)	1Q06-1Q07 growth (%)
Windows CE	3 184 703	62.1	1 937 667	52.8	64.4
Research In Motion	928 239	18.1	929 883	25.3	-0.2
Palm OS	314 353	6.1	489 220	13.3	-35.7
Symbian	288 000	5.6	132 000	3.6	118.2
Linux	33 400	0.7	43 530	1.2	-23.3
Others	377 150	7.4	137 000	3.7	175.3
Total	**5 125 845**	**100.0**	**3 669 300**	**100.0**	**39.7**

Note: Excludes smartphones, such as Treo 750 and BlackBerry 81xx, but includes cellular PDAs such as BlackBerry 87xx.

Source: Gartner Dataquest (May 2007), *www.gartner.com/it/page.jsp?id=506328*.

As software-driven technology becomes more mobile, new markets are emerging to take advantage of new opportunities. The growing demand for mobile content – such as music, games, data and multimedia materials – is providing further incentives for innovation by software developers as well as by stakeholders in the telecommunications, electronics and media industries. In addition to rather conventional services such as ringtones, news, and games, higher levels of services such as financial services and mobile television are being brought to the market. As the market develops, numerous actors – including users – are taking part in various parts of a complex and changing value chain. Yankee Group estimate the mobile data service to reach USD 146 billion in 2009 and Portio Research predict mobile content market to be more than triple to USD 59 billion over the four years to 2009 (OECD, 2006a).

Mobility in enterprise and business

The rapid growth of wireless networks, mobile functionality and mobile devices are having impacts on business well beyond the ICT sector. Some of early economic analysis shows that mobility can have major effects on labour productivity at firm level (OECD, 2004b). Consequently, it is not surprising that enterprises are putting increased priority on "mobilising" their workforce in order to make them more effective and available.

According to an IDC forecast, the number of mobile workers in the US will increase from 105 million in 2006 to over 120 million by the end of 2011 and for Europe, the number is expected to grow from 84 million to 91 million over the same period. IDC estimates that the mobile enterprise application market will grow from USD 1.2 billion in 2005 to 3.5 billion in 2010, representing a compounded annual growth rate of 23%.[20]

As businesses look at an increasingly mobile workforce, they are seeking ways to permit secure access to corporate information over mobile devices. Device-based security (such as "device wipe" or locking and encryption of data) is among the most important features that a mobile security solution must offer. Besides maintaining secure access, the mobile enterprise applications deliver contents beyond email; the ability to view documents, participate in video conference and access company software tools. Applications such as multi-party conference calling, dedicated voice key, push-to-talk over cellular and internet call (VoIP over WLAN) are being used to reach mobile workers.

The expansion of "M-commerce" has generated growing demand for mobile device security software from both businesses and consumers. According to a forecast from IDC, worldwide mobile security license and maintenance revenue exceeds USD 200 million in 2007 and will continue to grow at a healthy annual growth rate through the forecast period. In the future, features like mobile firewall, mobile VPNs (virtual private networks), and mobile antivirus protection are expected to gain increased importance for people seeking to secure use of expanded mobile functionality (IDC, 2007)

Demand and implication for software sector

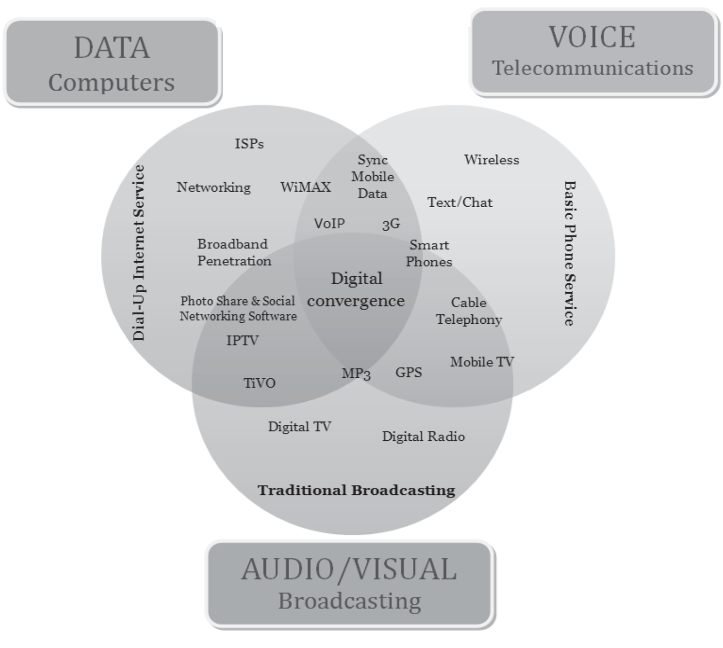
The increasingly diverse and geographically disbursed computing environment poses a number of opportunities as well as challenges for software developers. One of the challenges is finding ways to support mobility (Sousa, 2002). Market demand is encouraging developers to find new ways to enable users to take advantage of local capabilities and resources in a given environment, while also providing access to external users, devices and resources that operate in change conditions. This means that the software cannot be designed to rely on a fixed set of hardware devices; it must adapt itself to available devices. Often, the response is to aim for a software architecture that reduces the amount of device-specific development to the maximum extent possible. Software-as-a-service and cloud computing (discussed in Chapter 1) offer platform-independent mobile access, for example.

The same logic can apply to a user who may change context (*e.g.* when travelling with a laptop and PDA), with consequences for communication and system structure. This means that although the set of hardware devices is known, the structure of the communication software components must be able to adapt to itself to the user's context, which requires support from the software architecture (MacWilliams, 2004). Applications should be able to follow the user and move seamlessly between devices. Also, since mobile software interacts with software located at other network platforms, mobile software must be robust in the face of intermittent connectivity.

Summing up

Mobility has become a key software functionality and it is in demand across the various software markets. This characteristic is now featured across a broad range of business and consumer applications. The utility of this functionality has been widely recognised and its inclusion in software has become de rigor in many situations. Together with enhanced hardware capabilities, this realisation is promoting innovation for better integration of systems regardless of the geographical position. It is also related to technological convergence, whereby the development of mobility as a software functionality is associated with connectivity and multifunctionality in devices with data, voice and audio-visual applications (Figure 3.1).

Figure 3.1. Examples of convergence in data, voice and audio-visual technologies



Source: US Department of Commerce.

Interoperability[21]

The term interoperability is used in various contexts and with various meaning. According to one report of the International Organisation for Standardization, interoperability may be defined as "the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the users to have little or no knowledge of the unique characteristics of those units".[22] The EU Software Directive 1992 defines interoperability between computing components generally to mean "the ability to exchange information and mutually to use the information which has been exchanged". The US E-Government Act of 2002 defines interoperability as "the ability of different operating and software systems, applications, and services to communicate and exchange data in an accurate, effective, and consistent manner".

The key to interoperability can be found in the ability of outsiders to have access to the structure of the technical interfaces of software to which a connection is desired. The Internet is among the most obvious vehicles for interoperability, whereby any software can connect and exchange data as long as it adheres to the key protocols.

In considering the different aspects of interoperability, one might distinguish between two dimensions: vertical and horizontal interoperability (Weiler, 2005). Vertical interoperability can be seen as access between applications providers and platform owners. Application program interfaces (APIs) enable interaction between their software products and depend on availability of interface information to application providers for a given platform. Achieving vertical interoperability is a complex task that is a result of co-operation between software developers. This is because vertically related products are highly integrated and may be provided by many developers. For example, access to the Internet involves several layers of vertically related products and applications; hardware, operating software, and application platforms such as a Web browser.

Horizontal interoperability can be seen as the ability of a developer to write an application that can run on different operating systems, platforms or other computing environment. Horizontal interoperability may be established by writing an application that runs on multiple platforms by calling the respective API set of a platform when running on it. Horizontal interoperability might also include different platforms implementing common protocols that can be used by applications or other platforms to communicate them in a consistent manner. For example, many components of the Internet communication protocol standards such as TCP/IP, the World Wide Web and domain names, have been developed through consensus process and are now accepted by the marketplace.

Even though there are competing definitions of interoperability, their main goal is rather clear: to enable heterogeneous software products and services that are components of information and communication technology systems to work harmoniously. This can promote an increase in user confidence, value and choice, as well as competition among providers, by removing technical impediments to the use of products and services from various vendors. With interoperable products and services, the user may not need to choose a specific technology or change equipment as often. Depending on the context, interoperability may also help foster the acceptance, success and penetration of new technologies. Ultimately, users decide whether they value interoperability in a particular situation and the degree to which they want it, particularly if there are trade-offs with other features (security, reliability, cost, etc.).

The demand for interoperability

The software industry is undergoing an evolution where systems are frequently built by assembling heterogeneous, off-the-shelf or proprietary software or mixed and by integrating these with external systems. As this evolution continues, developers and some users are learning new techniques and processes for building systems in this new mixed environment; this can requires a deep knowledge of integration and interoperability issues. An increase in technical heterogeneity drives a greater demand for interoperability between different components as stakeholders try to find better ways to innovate and improve their ICT performance.

From the supplier's perspective, the heterogeneity leads to increased market demand for their solutions to be capable of successfully working in a mixed ICT environment. Businesses, for example, are looking for any opportunity to reduce costs. Developers no longer enjoy the luxury of developing every component nor using an array of adapters for every unique requirement. Instead, applications that are built using interoperable techniques can speed up deployment and reduce cost. Interoperability increases firm revenue and efficiency of business process by allowing system synergies to be exploited and more cost effective technologies to be utilised.

The ability to deliver a product to interoperate with other products or services can have an impact on market demand for a product. In many cases, market forces and competition have provided incentives for developers to make products that are interoperable. Developers that fail to adhere to widely-adopted industry standards or fail to provide a minimum degree of access to their products may expose themselves to risk of losing competitive edge over those firms whose products do implement such standards or provide access for external developers. This risk can potentially be compensated in cases where the developer is able to bring into the market an alternative product of significantly higher quality (*e.g.* in cases where the developer introduces a breakthrough innovation).

Interoperability may also have economic consequences such as network externalities. If competitors' products are not interoperable due to causes such as lack of co-ordination, then there can be an exacerbation of market consequences (either increased market share or failure). At the same time, the substitutability achieved by interoperability can imply larger market size which may result in lower unit costs and prospects for better prices for consumers.

The need for interoperability in other sectors

Public sector and business activities rely heavily on information technologies. An important aspect of software in ICT intensive sectors (*e.g.* in aerospace, mobile telephony, petroleum, pharmaceutical and automotive) lies in its ever increasing complexity and fast rate of technology evolution. Individual suppliers are increasingly finding that collaboration is a critical dimension in meeting the demands driven by ever evolving business environment. It is virtually impossible for even the biggest and leading companies to supply the sophisticated functions needed by all potential users of its software. This leads to much emphasis in achieving software and data interoperability as the reach of particular software products extends from industry to industry and across various applications. Thus, interoperability can have a variety of implications for a modern economy.

There can be a variety of benefits from establishing interoperability Braunschweig (2005) provides an illustrative list:

- easy, cost-effective and error-free data transfer between applications,
- easy expansion of functionality in application software,
- avoidance of vendor lock-in,
- support to quality assurance process, and
- trigger for innovation.

At the same time, promotion of interoperability may also entail potential costs such as monetary costs to developers, possible security risks or undue constraints on innovation or functionality.

Government also influences market demand for interoperability, both as a customer and a policy maker. As customers, governments increasingly rely on ICT systems to carry out their respective missions. In procurement of systems and services for various operations, governments often consider interoperability as one selection criterion.[23] In part this can be driven by operational concerns. From a policy perspective, governments sometimes address interoperability concerns from the perspective of promotion of innovation or competition.

For example, EU policy initiatives have brought interoperability to centre-stage of the EU ICT governance framework. The IDABC (Interoperable Delivery of Pan-European e-Government Services to Public Administrations, Businesses and Citizens) programme of the EU commission is one example of a measure intended to address such issues.[24] Another illustration is the e-GIF initiatives (e-Government Interoperability Framework) in the United Kingdom, which sets out the government's technical policies and standards for achieving interoperability and information system coherence across the public sector (*www.govtalk.gov.uk*). Other public-private initiatives focus on information exchange, such as the National High Performance Computing and Communications (HPCC) Software Exchange in the United States, which serves as a web-based resource for "promoting software sharing and reuse within and the HPCC community" including with respect to interoperability.[25] In this regard, promotion of interoperability for public use (*e.g.* in relation health or safety systems) or general welfare should be distinguished from policies that may distort the functioning of markets.

Achievement of interoperability

Software interoperability can be achieved in various complementary ways. The most common approaches being adopted by ICT companies include: *i)* industry-community partnership and collaboration; *ii)* product design and testing; *iii)* sharing of technology and access to intellectual property (IP); and *iv)* implementation of technology standards (BSA, 2007; Gasser and Palfrey, 2007).

- *Standards implementation*. Implementation of existing technology standards in products and services plays an important role in achieving interoperability by providing a stable technical solution to a common problem. There are various types of standards falling into two broad categories: *i)* "proprietary" standards developed and maintained by single vendor or a group of vendors; and *ii)* "open" standards which are developed through an open, voluntary, consensus-based process and publicly available to any interested party.

- *Industry-community partnership.* Industry and community partners – and sometimes competitors – collaborate at the domestic or international levels to share technical information with the purpose to define or develop a way for their products or services to work together. This work could lead to the development of an industry standards through a formal standardisation body(*de jure*) or through general usage in the market place (*de facto*).
- *Product design and testing.* In response to either the common standard established by industry-community or customer demand, developers sometimes move to elaborate interoperability through expanded product development activities throughout the design and testing phase of products.
- *Sharing common technology and access to intellectual property.* The adoption of shared technology or intellectual property (*i.e.* through royalty or royalty-free licensing) may reduce the variability in the interface between different products and thus facilitate development of interoperable products in a quick and cost-effective manner.

The industry and government responses to the demand

Software interoperability has improved in the last decades despite the fact that complexity and variety of systems and new technologies have increased significantly. Major software industry firms are co-operating at an unprecedented level to align their technologies so that their products interoperate. In addition, there is increased engagement on the part of various stakeholders including ICT firms (*e.g.* network and systems management vendors), users and standards bodies. Through collaboration among interested parties, significant progress in software interoperability has been accomplished in recent years.

Collective efforts

Commercial software companies collaborate with other firms and also actively engage with broad-based standards bodies such as the World Wide Web Consortium (W3C), the Internet Engineering Task Force (IETF), the Web Services Interoperability Organization (WS-I), the European Computer Manufacturing Association (ECMA) and others (Box 3.1). They have been working together to address interoperability issues through standards development in technical committees of various organisations. One recent illustration of actions to promote interoperability is embodied in a group of standards based on the internet format XML (eXtensible Markup Language). XML is an open standard maintained by the W3C and many leading commercial software firms have contributed in the development of the basic XML architecture.

Box 3.1. Software-related bodies

The World Wide Web Consortium (W3C) is an international consortium established in 1994. W3C's main mission is to develop Web standards and guidelines that ensure long-term growth for the Web. Since 1994, W3C has published more than 110 such standards, called W3C Recommendations. W3C has over 400 member organisations, including vendors of technology products and services, and content providers, from more than 40 countries (*www.w3.org*).

ECMA International is an industry association founded in 1961 and dedicated to the standardisation of ICT and consumer electronics. The main activities of ECMA are to develop, in co-operation with the appropriate national, European and international organisations, Standards and Technical Reports in order to facilitate and standardise the use of ICT and consumer electronics. So far, ECMA has published more than 370 ECMA standards and 90 technical reports, more than two-thirds of which have also been adopted as International Standards and/or Technical Reports (*www.ecma-international.org*).

European Information & Communications Technology Industry Association (EICTA). Formed in 1999, then in 2001 it merged activities with EACEM (the European Association of Consumer Electronics Manufacturers). EICTA set up an Interoperability Task Force in September 2003, and produced a "white paper" on interoperability (EICTA, 2004).

The **Organization for the Advancement of Structured Information Standards (OASIS)** is a non-profit consortium, formed in 1993. OASIS acts as a driving unit to develop, and adopt standards of e-business and covers numerous areas including Web Services and e-Commerce (*www.oasis-open.org*).

The **Accessibility Interoperability Alliance (AIA)** is a global engineering collaboration between assistive technology vendors, IT companies, Web, test, and tool companies, and key NGOs. It is focused on addressing the complexity of developing accessible products and improving interoperability and stimulating the development of increased innovation around accessibility and more consumer choice (*www.accessinteropalliance.org*).

Firm level efforts

Software developers are putting much emphasis on interoperability. For example, in early 2008, Microsoft announced four principles regarding interoperability: "ensuring open connection, enhancing support for industry standards, promoting data portability, and fostering more open engagement with customers and the industry, including the open source community" (*www.microsoft.com/interop/principles*). In building on these four principles, Microsoft will rely on inputs from the Interoperability Executive Customer Council, consisting mainly of chief information officers or chief technology officers of large enterprises and government agencies. Microsoft expects this initiative will make it easier and less costly for developers to create software that works smoothly with its current products. Also, Microsoft opened more than 50 000 pages of documentation about interoperability and APIs to the public.

IBM also announced recently that it is collaborating with nine business partners to help healthcare providers, clinics and hospitals improve productivity, increase quality and reduce costs through the use of Service Oriented Architecture (SOA). These partners are working to develop their latest healthcare applications using the IBM SOA Foundation and supporting a set of open technology and industry standards.

Government-wide efforts

A European Parliament and Council Decision (2004/387/EC) promotes the Interoperable Delivery of Pan-European e-Government Services to Public Administrations, Businesses and Citizens (IDABC). The IDABC programme was launched on January 2005 for a period lasting until the end of 2009 in order to address problems that arise from ICT systems not being interoperable for both technical and organisational reasons. Interoperability is key to all of IDABC's horizontal e-government-related activities. It is expected that an integrated approach to interoperability will help to guide and support the development of electronic services. Activities to support interoperability have already been undertaken under IDABC's predecessor IDA II. These include the publication of the European Interoperability Framework (EIF 1.0), recommendations for promoting open document formats and an XML clearinghouse feasibility study (*www.ec.europa.eu/idabc*). In addition to these efforts, the first *eGOV INTEROP Conference* was held in 2005 as a major means for confronting the progress of the various initiatives relating to e-government interoperability in Europe. The conference focused on all aspects of interoperability in e-government strategies, both from technical as well as from semantic, organisational and socio-economic perspectives, and discusses the progress of the various European initiatives in the field (*www.egovinterop.net*).

Japan announced new interoperability guidelines in 2007 (METI, 2007a), which encourage Japanese government ministries and agencies to solicit bids from software vendors whose products support internationally recognised open standards. The Japanese government expects this new interoperability framework will propel healthy competition and open up more opportunities for small and medium-size companies in Japan. Japan has made further efforts to promote interoperability of software by publishing the "Technical Reference Model for e-Government System" at the end of 2008, which includes examples of recommended open standards for government's information system.

Issues surrounding interoperability

While many stakeholders – industry, government, and consumers – have agreed upon the need and benefits interoperability could bring, the scope, actual implementation or technical barriers that hinder it, remain somewhat controversial. One of the main controversies concerns the distinction between "open standards" and "open source". Open standards are vetted through an open process and can be seen as a set of rules and specifications that are based on the consolidated results of science, technology and experience, and aimed at the promotion of optimum community benefits (EICTA, 2004). While, open source refers to software in which the source code is available to the public for use and modification from its original design. Open source software may or may not implement open standards.

Some public authorities have put considerable effort into the promotion of interoperability including initiatives in relation to open standards. In the European Union, for example, a programme for implementation of interoperability known as IDABC (Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens) aims to facilitate the collaboration of public administrations in 27 different member states and in EU institutions. Following the outlines of an eEurope Action Plan, an IDABC decision (Decision 2004/387/EC) was prepared "based on a framework of common principles and rules, as well as, on the agreement on open standards and interfaces for the implementation of interoperability between systems, applications, business processes and actors producing or using eGovernment services."[26]

At the same time, some stakeholders stress that care should be taken to ensure technological neutrality and choice are maintained, in particular with respect to government actions in support of interoperability.[27] This can include an objective of seeking out and evaluating "best of breed" solutions based on the best value for the money, support, interoperability and compatibility (Lueders, 2005). A key concern is to avoid stifling potential innovative responses that may not yet be known to the public. Thus, there can be advantages in aiming for promotion of interoperability while maintaining an objective of technological neutrality in policy (see Box 3.2 on US policy regarding technological neutrality in public procurement).

Box 3.2. United States government technology neutral policy in public procurement

The United States government software procurement policies rely on the principle of intentional technology and vendor neutrality. The agency implementation should be similarly neutral, to the maximum extent practicable. According to the official guidance all IT investment decisions, including software, must be made consistent with the enterprise architecture and the Federal Enterprise Architecture. In addition, agencies need also to take into account "…*the total cost of ownership including lifecycle maintenance costs, the costs associated with risk issues, including security and privacy of data, and the costs of ensuring security of the IT system itself.*" These policies apply to acquisitions of all software, whether it is proprietary or open source software.

The official policy statements are guided by the Office of Management and Budget (OMB) and by the Federal Acquisition Regulation (FAR). They were also outlined in Memorandum M-04-16 on software acquisition (see *www.whitehouse.gov/omb/memoranda/fy04/m04-16.html*).

Summing up

The scope of interoperability extends across the ICT sector and beyond. It is not just a single company issue, nor sector specific, nor country specific, but global. Although the benefit of interoperability is generally agreed and significant improvements have been achieved in recent years, there remain different views for its future evolution such as trade-offs with other functionalities and cost. Substantial progress towards interoperability, for example in the area of cloud computing and web services, is being achieved through market mechanisms. At the same time, further promotion of interoperability is being undertaken through various channels including private sector, government and multi-stakeholder initiatives. Success in this regard, may also have benefits for other types of software functionality such as mobility.

Accessibility

In view of the wide-ranging – and expanding – role of software in social, civic and economic life, software developers have been challenged to fight exclusion and ensure accessibility on a technical level for as much of society as possible.[28] The impetus for innovation in software accessibility is coming from a variety of angles including, among others, consumer demand, government mandates and procurement[29], international standards, and the corporate social responsibility of developers. Changes in demographics, including concerns related to population aging, as well as the opening of new technological possibilities (*e.g.* for expanded wireless communication) have raised awareness of the expanding needs and the options to address those needs (Reed *et al.*, 2004). Significant progress has been made, but on-going technological change means that software accessibility issues will need on-going attention from developers and other stakeholders.

Definitions and benefits

Software accessibility can be seen as an issue centred on the possibilities for human-computer interaction. The International Organization for Standardization (ISO) defines accessibility as *"usability of a product, service, environment or facility by people with the widest range of capabilities."* The ISO definition can be compared to the definition from American National Standards Institute/Human Factors and Ergonomics Society (ANSI/HFES): "The set of properties that allows a product, service or facility to be used by people with a wide range of capabilities, either directly or in conjunction with assistive technology. Although "accessibility" typically addresses users who have a disability, this concept is not limited to disability issues" (Gulliksen, 2004). In addition, ISO 9241-171 (Ergonomics of human-system interaction) provides guidance on the design of software to achieve as high a level of accessibility and interoperability as possible.

One of the primary objectives of software accessibility is to provide interfaces that may be used or adjusted to meet the needs of a very broad range of people including those with disabilities[30] such as:

- visual: low vision, lack of colour perception, blindness.
- hearing: hearing loss, deafness.
- mobility: restricted movement or control of arms, hands and figures.
- learning or cognitive impairment (*e.g.* dyslexia).

Accessible software can benefit first and foremost those individuals dealing with such impairments. Yet, experience has shown that the potential benefits from improved accessibility can extend far beyond this population, contributing to general increases in productivity, reduced mental and physical stress, or economies in cost of training. For example, customizable fonts and colour may enable users to pick settings that reduce eye strain and display more effectively in particular operating conditions. Keyboard navigation functions may enable users to move faster than using a mouse. Having alternative means of providing operator input may lower the risk of repetitive stress injury. Today the demand for accessible technology is expanding, as more people with mild or moderate disabilities, or those who are aging, are realising the many ways accessible technology can help them work, learn or play -more effectively.

Innovative technologies

The drive for accessibility plays an important role in promoting technological innovation and diffusion and exploitation of recent technologies and design strategies. The application of these enhanced approaches may open opportunities for organisations and businesses to reach new customers and markets or deliver more effective services.

In some cases, accessibility is facilitated through add-on assistive technology such as software or hardware that is used to increase, maintain, or assist the functional capabilities of individuals with disabilities. This technology offers alternative ways to access the contents on the monitor and to issue commands. It can be any technique that assists people in removing or reducing technical barriers and enhancing their activities. Examples of assistive technologies available are:

- screen reader software for those with visual disabilities, which operates by detecting and reading text displayed on the computer monitor using text-to-speech synthesis.

- voice recognition software that allows a person to simulate typing on a keyboard or selecting with a mouse by speaking into the computer.
- screen magnification software that allows a low-vision user to read more easily information displayed on a monitor.
- comprehension software that allows a dyslexic or learning disabled person to see and hear text as it is manipulated on the computer screen.
- keyboard enhancements and accelerators such as sticky keys, repeat keys and bounce keys.
- software mouse simulators that allow users to move the mouse pointer by pressing keys on the numerical pad.

Government polices

In recent decades, governments in many OECD countries have increased the priority they place on this issue. Many have moved to encourage enhanced software accessibility via voluntary guidelines, awareness raising campaigns, procurement practices, legislation or standards. For example, in the late 1990s, the United States government promoted its e-accessibility policy (Box 3.3).

Box 3.3. United States government e-accessibility policy

In the United States the issue of accessibility for ICT is regulated by Section 255 of the 1996 Telecom Act and Section 508 of the Rehabilitation Act requiring Federal agencies to purchase electronic and information technology that is accessible to people with disabilities. The law applies to all Federal agencies when they develop, procure, maintain, or use electronic and information technology. These agencies must ensure that this technology is accessible to employees and members of the public with disabilities to the extent it does not impose a burden. Section 508 speaks to various means for disseminating information including computers, software, and electronic office equipment. It applies to – but is not solely focused on – federal government pages on the Internet or the World Wide Web.

The principal government agencies involved in e-accessibility issues are: the US Access Board, the General Services Administration (GSA), and the Department of Commerce (DOC). The Access Board is responsible for developing accessibility standards, including for ICT equipment. It is also responsible for creation of accessibility guidelines for telecommunications equipment (a process it initially completed in 1998) and to issue standards for electronic and information technology (a task it initially completed in 2000). In September 2006, the Access Board began to update the standards requirements of Section 508 of the Rehabilitation Act and Section 255 of the Telecommunications Act. To help achieve international harmonisation in e-accessibility standardisation, the Access Board invited representatives from Japan, the EC, Australia, and Canada to participate on its Telecommunications and Electronic and Information Technology Advisory Committee (TEITAC), which also includes representatives from US disability organisations, industry trade groups, and Federal agencies. The TEITAC completed its work this past spring. The standards developed by the Access Board were incorporated by the GSA into the Federal Acquisition Regulation in 2001, making them mandatory for all Federal agencies whenever they develop, procure, maintain, or use electronic and information technology.

In addition, all state governments that receive federal funding for assistive technology programmes are often required to follow Section 508 guidelines on e-accessibility.

Sources: www.jtc1access.org, www.eaccessibility-progress.eu/country-profiles, www.sapdesignguild.org/editions/edition9/policies2.asp, www.microsoft.com/enable/news/default.aspx.

The European Union provides another example. In 1999, the European Commission proposed an initiative called eEurope. The main purpose of this initiative was to bring the benefits of the information society to all Europeans. The Commission followed up this initiative in a communication of 2001 stating that one of the specific objectives of the eEurope Action Plan is to improve access to the Web for the 37 million people with disabilities in Europe as well as for the growing number of older persons. In 2005, the Commission adopted a further Communication on eAccessibility aiming to ensure that as many people as possible can fully participate in the Information Society. In particular, the initiatives encouraged the ICT sector and public bodies to better take into account the needs of the elderly and people with disabilities. Moreover, the Commission is promoting adoption of the World Wide Web Consortium's (W3C) *Web Content Accessibility Guidelines* for all public websites in European institutions and member states.

Japan, as well, has been actively involved in accessible ICT initiatives, contributing a number of proposals to the ISO and co-operating with other Asian countries. The Japan Electronics and Information Technology Association is responsible for developing industry standards that will foster a digital network society aimed at improving quality of life via ICT advancement. In 2003, the Japan Industrial Standard Committee (JISC) issued the standardisation policy addressing needs of the elderly and persons with disabilities, which includes establishment of a systematic standardisation scheme, prioritising of standardisation fields including via annual updates, and encouraging more international co-operation (Yoshida, 2006).

Korea also has taken a leading role in promoting ICT accessibility polices. Anti-Discrimination against and Remedies for Persons with Disabilities Act came into effect on April 2008 which include polices toward providing information access rights and reasonable accommodations in ICT. The Korean government has evaluated central and local government websites according to the Web Accessibility Standard since 2005, and expanded the website testing to 326 regional governments. Also, nearly 3 000 web designers and developers from public and private sector on web accessibility were educated since 2005 (*www.unescap.org/esid/meetings/ICT/PresentD.pdf*).

Guidelines and standards

There is a wide range of resources available to software developers seeking guidance in designing accessible software-user interfaces. A number of standards or guidelines have been published by international organisations, governments, non-governmental organisation and industry associations. Various supplementary resources are available from bodies such as ISO and W3C, private entities like Apple, IBM, Microsoft, the Mozilla Foundation and Sun, and academic institutions such as the Trace Centre at University of Wisconsin-Madison.[31] These include, for example, academic studies, design guidelines, training materials and tools. Co-operation among different stakeholders has also been active. The proposed US Software Accessibility standard was contributed to ISO in 2000, and served as a basis for ISO Technical Report 16071 – Guidance for Software Accessibility (*Reed et al.*, 2004). In Asia, the Japan-Korea-China Accessibility Design Committee was established in 2003 to promote standards co-operation among standards bodies.

Although there are many different guidelines that reference software accessibility, they tend to share some common features. For example, they generally address one or more categories of software accessibility issues (Kavcic, 2005):

- system and content, including object information and timing
- keyboard access and pointing devices
- sound and multimedia
- display
- verification of accessibility and documentation

Guidelines for Web accessibility are often also directly relevant to software accessibility, albeit with more specific treatment of the Internet dimension (Brewer, 2004). For example, the W3C Web Accessibility Initiative has delivered a series of guidelines (Box 3.4).

Box 3.4. W3C Web Accessibility Guidelines

Web Content Accessibility Guidelines (WCAG). WCAG 1.0 became a W3C Recommendation in 1999. Since then, many organisations and private sector firms have adopted the Guidelines. WCAG is composed of general guidelines and specific checkpoints against which conformity can be evaluated. The checkpoints are divided into three priority levels, addressing: *i)* absolute barriers, *ii)* significant barriers, and *iii)* additional accessibility support for disabled people. The development of WCAG 2.0 is now underway, taking into account extensive feedback on WCAG 1.0 and advancements in Web technology.

Authority Tool Accessibility Guideline (ATAG). This guideline became a W3C Recommendation in 2000. It incorporates guidelines on how to make software applications that are used to create Websites accessible to disabled people and ways to support the production of accessible Websites. It includes a checklist to ask software vendors when evaluating accessibility support in their Web authoring tools.

User Agent Accessibility Guidelines (UAAG). UAAG became a W3C Recommendation is 2002. It can be considered as complimentary to WCAG, since software used for accessing the Web also needs to be accessible. UAAG explains how to make browsers and media players accessible to disabled and how to make them interoperate smoothly with assistive technologies.

For more information see the W3C website: *www.w3.org/WAI.*

Company efforts

Many software firms are committed to making their products broadly accessible, including users with disabilities or impairments that occur with aging. Most of them look to accessibility best practices and standards defined by guidelines such as Section 508 of the US Rehabilitation Act and the W3C WAI. Several illustrative cases are presented below:

- Microsoft, for example, has publicly declared its commitment to accessibility with the adoption of its accessibility policy and increased its accessibility efforts in 1995. Since then, Microsoft has been developing innovative accessible technology solutions that meet the individual needs of people with a wide range of disabilities. Its leading products, such as Windows, Microsoft Office, and Internet Explorer, are all built with accessibility in mind. [32] For example, Microsoft invested more than three years of research to better understand the needs of people with a wide range of impairments that can affect their computer use when developing Window Vista. Major accessibility improvements include state-of-the-art speech recognition, enhanced magnification and text-to-speech capabilities, and the new Ease of Access Centre. Besides developing assistive technologies, the firm has also made efforts to raise awareness, support effective public policy, provide design guidance, and run accessibility resource centres to demonstrate available accessible technology solutions.

- IBM formed a worldwide accessibility centre with locations in the United States, Europe, Japan and Australia in 2000.[33] The centre aims to foster product accessibility by working toward the harmonisation of worldwide standards and applying research technologies to solve problems experienced by people with disabilities; it creates industry-focused solutions and generates accessibility awareness. As part of the IBM research organisation, the Accessibility Centre has a direct line to the scientists developing new technologies.
- In 2001, SAP established the Accessibility Competence Centre (ACC). The ACC is the focal point for the SAP Accessibility Programme which aims to ensure the firm meets or exceeds compliance with accessibility requirements outlined in legislative and industry standards.[34]

Summing up

There has been considerable progress in addressing software accessibility issues as highlighted in various examples above. However, technological change and progress mean that the work to enhance accessibility remains on-going and that new opportunities may arise to better response to accessibility needs. In recent decades, various institutions – public and private – have developed and joined efforts to ensure a continued focus on this work and to assist in the diffusion of best practices.

Reliability

With the world becoming more and more dependent on software, software failures can cause more than mere inconvenience. Today, software errors or failures can expose society to the risks of critical infrastructure failures, severe economic loss or even human fatalities. Where they occur, reliability problems can originate from a variety of sources ranging from poorly designed user interfaces to direct programming errors or improper implementation. In view of the increasing complexity of software, efforts to address reliability needs now engage the full range of stakeholders including developers, vendors, users and others.

Reliability is part of a cluster of inter-related concepts linked to the ability of users to have confidence that the software products they employ will work consistently and not in an unpredictable manner. Software experts apply a variety of terms and definitions in describing various perspectives on this cluster of issues, such as reliability, and quality assurance. These terms can be defined as follows:

- Software reliability: "The probability of failure-free software operation for a specified period of time in a specified environment." (ANSI, 1991 and Musa *et al.*, 1982, as cited in Lyu, 2007)
- Assured software: "Software that has been designed, developed, analysed and tested using processes, tools, and techniques that establish a level of confidence in its trustworthiness appropriate for its intended use." (CNSS, 2006)[35]

Dependability is another concept that is closely related to *reliability* (Box 3.5). Dependability is defined as "the trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers", which includes as special cases such attributes as reliability, availability, safety, security (IFIP, 1988). Hence, the term "dependability" can be generally regarded as a broader concept to "reliability." In addition, the concept of "dependability" puts emphasis on user's perspective, thus it becomes one of the factors that impact software innovation through user's needs for dependable system.

Box 3.5. Software dependability

Dependability is a broader concept than software reliability. According to Avizienis *et al.* (2004) "dependability is first introduced as a global concept that subsumes the usual attributes of reliability, availability, safety, integrity, maintainability, etc." The American National Standards Institute presents dependability as "a key decision factor in today's global business environment [that] affects product costs and process. [Dependability] is an inherent product design property influencing product performance. • Dependability is the collective term describing the availability performance of any simple to complex product. The factors influencing the availability performance of a product are the reliability and maintainability design characteristics and the maintenance support performance" (ANSI, 2003).

Avizienis *et al.* (2004) illustrates the relationship between "dependability", "reliability" and "security" in the following figure:

Dependability —	←Availability →	— Authorised actions	— Security
	←Reliability		
	←Safety		
	Confidentiality →		
	←Integrity →		
	←Maintainability		

There are several national actions to strengthen the dependability for software products. For example in Japan, Ministry of Economy Trade and Industry (METI) started several initiatives to promote dependability on software products including: publication of "Guidelines for Improving the Dependability of Information Systems" in 2006, and launching a study forum on dependability and security of software for an advanced information society from the autumn of 2008.

As Michael Cusumano – a leading software sector expert based at MIT – points out, successful software development is a creative process requiring a balance between establishment of enough structure to keep projects under control, but not so much as to stifle creativity and flexibility (Cusumano, 2004). Despite significant progress in software development approaches in recent decades, he notes that problems have persisted. For example, he points to observations that as many as 75 to 80% of software projects are typically late and over budget.

Concerns about reliability are heightened by user perceptions. A survey of business enterprises in Japan (Nikkei, 2003) found that 73% of system development projects fail to deliver as expected in one or more key dimensions specified as "quality", "cost" or "delivery". The respondents pointed to the increased costs or delays associated with these shortfalls.

To some extent, the problems of achieving reliability in software systems reflect the difficulties of programming. As Cusumano (2004, p. 132) notes, "Writing program algorithms is usually not a routine activity. It generally involves creativity and invention on some level, as well as problem solving and trial and error." Moreover, "In custom software projects, users often do not know what they want until they see part of the system in front of them." The complexity of the programs and the environment ensures that achievement of reliability is a clear challenge.

Attempts to improve software reliability can be applied in different parts of the development process, which might be frames broadly as including requirement specification, design, programming, testing, and run time evaluation. Donzelli *et al.* (2006) note that dependability requirements are particularly difficult to deal with because they cover many different aspects of a system at the same time. Moreover, such requirements are deeply rooted in the specific context. Different stakeholders will focus on different attributes such as availability, catastrophic failure avoidance, and deliberate-intrusion prevention, and also differ in the definitions of these attributed. In some areas such as defence or nuclear energy or automotive safety, zero failures may be tolerated. In others, the objective may be 'reasonably low failure rates". Given these considerations, it is not surprising that Donzelli *et al.* note that erroneous or omitted requirements are often indicated as the main reasons for project failures.[36] A similar view was echoed by a survey in Japan of firms purchasing software systems; among firms that experienced failures of projects to deliver expected results, the leading cause – cited by 36% of respondents – concerned requirement definitions (Nikkei, 2003).[37]

In recognising the importance of dependability of software, Japan has taken extensive measures to facilitate software dependability including publishing "Guidelines for Improving the Dependability of Information Systems". (Box 3.6) In Europe, an initiative called SecurIST has been launched in order to have a clear European level view of the strategic opportunities, strengths, weakness, and threats in the area of ICT security and dependability (*www.ist-securist.org*).

Box 3.6. Japanese government policy measures for improving dependability

In recent years, the Ministry of Economy, Trade and Industry (METI) of Japan has taken various measures to improve the reliability of information systems. Most significantly, in 2006 METI introduced guidelines for improving the dependability of information systems that cover the entire scope, *i.e.* from design and development to maintenance and management. One of the main objectives of the guidelines was to promote reliability among IT industries. Measures include preparation of model agreements by private entities and benchmarking methods. In 2007 METI carried out follow-up with over 90 companies from various sectors to improve the reliability of information systems. In January 2009, METI revised the guidelines to be more useful and effective for users and vendors.

In response to the guidelines, the Japan Information Technology Services Industry Association (JISA) has conducted a survey in order to collect best practices on the use of management metrics in software reliability among industries. In addition a task force organised by METI including experts from academia and industries was formed to reach a common understanding between users and vendors and these reports were issued as "Model Trade and Contracts for Information Systems" in 2007 and 2008.

Sources:
www.meti.go.jp/english/information/downloadfiles/PressRelease/InfoSystem.pdf
www.meti.go.jp/policy/it_policy/keiyaku/index.html

While a number of approaches and tools have been developed to reduce the likelihood – in advance – of reliability issues arising.[38] Developers employ design and code reviews to maintain quality controls as projects advance. Software reliability can be enhanced during the process through techniques such as fault prevention, fault tolerance, fault removal and fault forecasting[39] (Avizienis *et al.*, 2001). However, reliability issues invariably arise and developers rely on applied testing of software as a primary technique for detecting them so they can be tackled. Developers employ such testing in an on-going manner during the development process with respect to such dimensions as the functioning and the stability of software. This applies to both the individual modules being developed and the interaction between modules. Issues are then addressed as they come to light through the testing procedures.

There can be numerous ways to conduct tests of the reliability of software using both human testers and automated tests. Two traditional approaches are black box testing and white box testing. Black box testing treats the software as a black-box and is implemented without requiring knowledge or understanding of internal structure of the software. White box testing is used when the tester has access to the internal data structures, code and algorithms and takes these factors into account. Under one common practice in software testing, an independent group of testers may review the product after it is developed and before it is delivered to the customer. Alpha testing is performed by potential external users or by an independent test team at the developers' site. Then follows beta testing during which the so-called "beta version" is released to an independent group of users. Sometimes, the beta versions are made available to the public in order to increase the volume of feedback.

The challenges in software reliability stem in part from the complexity and difficulty of analysing software applications in various domains. Thus, it is not surprising that the testing procedures can be extremely labour-intensive. In view of the complexity and difficulty in anticipating all the potential bugs, in some cases developers may be reluctant to spend too much effort in this area; the cost-effectiveness of additional effort may not be clear (Lyu, 2007). That is, the cost of reliability can be a direct function of the cost of testing and it can be very expensive. When the product delivery schedule is tight, reliability may receive lower priority. In addition, although many companies are collecting failure data, it is often the case that they will not share the data or experiences, which makes comparing or benchmarking reliability results difficult.

On the other hand, as Cusumano (2004, p. 177) points out, failure to address reliability issues (and quality issues more broadly) at the testing stage can entail massive costs due to product recalls and service provision. In the case of critical systems (such as safety-related systems), the costs of failure may be even larger. Hence, many developers invest enormous amounts in efforts to ensure an appropriate degree of reliability. While the relationship between reliability enhancement and this investment is not necessarily one-to-one (*i.e.* because problems vary in the degree of difficulty they pose), it can help. As Cusumano advises software entrepreneurs, "you can never do too much testing."

Many developers have mechanisms to solicit feedback from users even after release of the software, which provide input for revisions aimed at improving the reliability of a software product in future updates and releases. In some cases, failure reports are generated automatically via built-in routines while in others there are easy access links to facilitate reporting of bugs.

Box 3.7. User participation in the development of software functionality

In many sectors, the engagement of users in product development processes is a practice that is still only in its infancy. In the software sector, however, user-driven innovation is already widespread and continues to evolve. Indeed, the software industry can be considered as being among those on the leading edge of this type of development.

The active participation of users in the development of software is facilitated by the very nature of software as a digital, intangible product. For example, these characteristics enable real-time, global transmittal of content, insertion of incremental changes in existing products and automation of certain user input provision (*e.g.* software failure reports). Users are contributing concepts and even software code that goes beyond what otherwise would have been available. As a consequence, many software developers are able to produce products with a competitive edge. The types of advantage may vary by product, but can potentially include better time-to-market, quality, cost or functionality, among other characteristics, in comparison with products developed using more traditional approaches.

The manner of engaging user input may vary widely, from passive monitoring of usage and behaviour to more active engagement at the initiative of either the users or software developers. The resulting input may be handled in a closed proprietary manner between the parties or shared broadly via more open approaches. One common approach to encouragement of this type of user engagement is for developers to circulate alpha or beta versions of software, in some cases via the internet; leading users or early-adopters of the provisional technology can then make a contribution via feedback or other input. Another approach is "crowdsourcing", whereby problems are posed via closed networks or the Internet to various communities in an open manner in order to draw on a wide range of expertise.

OECD (2007e) presents the concept of the "participative web", which is based on "an Internet increasingly influenced by intelligent web services that empower users to contribute to developing, rating, collaborating and distributing content via the Internet. This concept is quite relevant in the case of software as users increasingly engage via the Internet in the process of software development in a variety of ways such as articulating their need and suggestions for enhanced functionality, creating their own applications via web-based platforms (discussed in Chapter 1), and contributing specific elements as input to external software developers (*e.g.* building on existing applications).

New web software tools enable users, even individual non-technical users, to contribute actively. Often, the user-driven input is facilitated by creative approaches to communication among the various stakeholders (*e.g.* using web-based platforms) and among various software applications via open web standards and interfaces. Thus, the so-called "collective intelligence" of the Internet users can be a creative force in software development including such well-known products as the Linux operating system, Firefox browser or Apache server software.

Incentives for users to engage in innovation processes operate with respect to businesses and individuals. For businesses, the incentives may be direct remuneration (*e.g.* as when they sell applications via platforms such as software.com) or indirect benefits through improved software functionality that enhances their own products or processes (*i.e.* leading to increased revenues or decreased costs). For individuals, the incentives may be monetary (*e.g.* payments, licensing fees or voluntary contributions) or quasi-non-monetary (*e.g.* recognition, enhancement of reputation or experience gained). In the literature, this latter type of incentive has sometimes been cited as "intrinsic motivation" (Bitzer *et al.*, 2007).

Summing up

Software reliability can be viewed from various perspectives and must be considered in context. Software developers have learned a lot over the last few decades about approaches to enhance reliability of software in line with expectations and market demand, and here as well there have been innovations (*e.g.* in methods). Nevertheless, reliability and related software qualities can be expensive and there can be tradeoffs between enhanced reliability and the cost of software.

Notes

1. This sometimes happens with open source systems.
2. For example, see Parker and Van Alstyne (2008).
3. The choice of functionalities for examination in this section was based on the priority interests of the OECD member countries as specified in the scoping document for the project.
4. This section draws on a paper prepared by Nick Mansfield, consultant to the OECD, under the supervision of the Secretariat for the OECD Committee on Information, Computer and Communications Policy (ICCP), the Working Party on Information Security and Privacy (WPISP), and in consultation with the Secretariat for the Committee on Industry, Innovation and Entrepreneurship (CIIE).
5. "Personal data should be protected by reasonable security safeguards against such risks as loss or unauthorised access, destruction, use, modification or disclosure of data."
6. "Ethics: participants should respect the legitimate interests of others […] Democracy: the security of information systems and networks should be compatible with essential values of a democratic society. Security should be implemented in a manner consistent with the values recognised by democratic societies including […] the confidentiality of information and communication, the appropriate protection of personal information, openness and transparency."
7. See Principle 6, "Risk Assessment" in the *OECD Guidelines for the Security of Information Systems and Networks* (OECD, 2002).
8. Professor of Computer Science, Purdue University.
9. See OECD (2008a), Section 2, "Government Authorities and Agencies".
10. BIAC/ICC (2003, 2004).
11. The data controller is the person who "is competent to decide about the contents and use of personal data" (OECD Privacy Guidelines). It is often the operator of the software that processes personal data.
12. The difference in the nature of the principles set forth in the OECD Security Guidelines and Privacy Guidelines reflect this fundamental difference between security and privacy.
13. Dewar and Schonberg (2008).
14. User perceptions of security issues in this regard are sometimes reported in the media. For example, see articles from *Byte and Switch* (7 April 2008) *www.byteandswitch.com/document.asp?doc_id=150418* and *TechNewsWorld* (27 February 2007) *www.technewsworld.com/story/55971.html* (both articles last accessed on 29 September 2008).

15. See, for example, Microsoft's *Privacy Guidelines for Developing Software Products and Services*, available at: *www.microsoft.com/downloads/details.aspx?FamilyID=c48cf80f-6e87-48f5-83ec-a18d1ad2fc1f&displaylang=en* (last accessed on 29 September 2008).

16. Niemelä and Latvakoski (2004) offer a more technical presentation of three types of mobility:

    1. Actual mobility: an extension in the capability of an autonomous software agent that enables the dynamic transfer of code and data towards nodes containing relevant resources. Exploitation of actual agent mobility can save network bandwidth and increase reliability and efficiency of the execution;

    2. Virtual mobility: the ability to recognise and function in an environment with multiple, networked options for execution; and

    3. Physical mobility: the possibility for mobile and wireless computing devices to connect to the Internet from dynamically changing access points.

17. 3G refers to the third generation of standards for mobile phones developed under the auspices of the International Telecommunication Union.

18. Other wireless technologies operate in a personal area for very local mobility. These include RFID, NFC and Bluetooth technologies, among others.

19. IDC (2008), *Worldwide Converged Mobile Device 2008–2012 Forecast and Analysis: A Category Comes of Age*, *http://idc.com/getdoc.jsp?containerId=211431*.

20. Information from the IDC website, downloaded in June 2008 from the following location, *www.idc.com/getdoc.jsp?containerId=prUS20491506*.

21. The concept of reliability is applied here with respect to software primarily in an operational sense. There are other dimensions such as programmatic interoperability, which describes the manner in which organisations work together to achieve interoperability. More information on these aspects can be found, for example, on the website of the Software Engineering Institute at Carnegie Mellon University: *www.sei.cmu.edu* (last accessed on 8 August 2008).

22. ISO (2003), available at *http://old.jtc1sc36.org/doc/36N0646.pdf.*

23. In procurement, interoperability is sometimes considered in the context of technological neutrality and the functioning of markets. For a discussion, on these topics (*e.g.* in relation to consumer choice and risks of constraints on innovation), see Tsilas (2007).

24. IDBAC is engaged in a number of initiatives aimed to facilitate voluntary co-operation among EU member states to promote pan-European eGovernment interoperability. IDABC does not set the EC's official interoperability and standards policy.
    See *http://ec.europa.eu/idabc/* (last accessed on 29 September 2008).

25. For more information on the HPCC software exchange, see
    www.dlib.org/dlib/may98/browne/05browne.html (last accessed on 30 September 2008).

26. More on these points can be found on the following IDABC web pages:
    http://ec.europa.eu/idabc/en/home and *http://ec.europa.eu/idabc/en/chapter/5883*.
    It is notable that one of the IDABC activities to achieve interoperability is "Support for Open Standards and Open Source Software", *http://ec.europa.eu/idabc/en/document/5313/5883*.
    These webpages were last accessed on 29 August 2008.

27. For example, see Tsilas (2007) who argues that key elements in promoting interoperability should include: *1)* protecting intellectual property; *2)* avoiding technology mandates which stifle innovation and stunt economic growth and *3)* promoting choice and technological neutrality in their procurement decisions and regulations. In another example, F. M. Buono

and M. Sieverding argue for choice and technological neutrality in government in a web posting and newspaper article available at: *www.metrocorpcounsel.com/current.php?artType=view&artMonth=February&artYear=2008&EntryNo=7853* (last accessed on 29 September 2008).

28. The focus in this section is primarily on accessibility as a human-computer interaction issue that deals with the ability to use computer systems. Some observers interpret this concept more broadly as a social issue of access to ICT or a quest to close the so-called digital divide (*e.g.* see *www.internetworldstats.com/links10.htm*, last accessed on 8 August 2008).

29. For example, see TEITAC (2008).

30. More information on the relationship of impairments, technology and software accessibility can be found on the website of the Royal National Institute of Blind People at: *www.rnib.org.uk/xpedio/groups/public/documents/PublicWebsite/public_sactypes.hcsp*.

31. For example, see the Trace Center website: *http://trace.wisc.edu.*

32. More information is available on the Microsoft website at: *www.microsoft.com/enable/microsoft.*

33. More information is available on the IBM website at: *www-03.ibm.com/able/accessibility_services/index.html*.

34. For further information, please see the SAP website at: *www.sap.com/platform/netweaver/standardssupport/accessibility.epx* .

35. A related issue concerns information assurance, a concept generally linked to security and integrity of data. For a discussion of the relationship to quality, see: Voas and Wilbanks (2008).

36. One of the approaches that address software dependability is the Unified Model of Dependability, which supports stakeholders in formulating their requirements and mapping them effectively (Donzelli *et al.*, 2006).

37. This is based on the sample of 498 companies that provided valid responses.

38. For example, according to IEEE (2004), IEEE Standard 982.1-1988 – *IEEE Standard Dictionary of Measures to Produce Reliable Software* – provides a set of measures for evaluating the reliability of a software product and for obtaining early forecasts of the reliability of a product under development.

39. However, it remains difficult to measure and model software reliability (Pan, 1999).

References

ANSI (1991), "Standard Glossary of Software Engineering Terminology", STD-729-1991, ANSI/IEEE, 1991, as cited in Pan, Jiantao (1999).

ANSI (2003), Dependability Management – Part 1: Dependability Management Systems, American National Standards Institute, IEC 60300-1 Ed. 2.0.

Avizienis, A., J.C. Laprie and B. Randell (2001), "Fundamental Concepts of Dependability", technical report, LAAS-Newcastle University-UCLA.

Avizienis, A., J.C. Laprie, B. Randell and C. Landwehr, C. (2004), "Basic Concepts and Taxonomy of Dependable and Secure Computing", *IEEE Transactions on Dependable and Secure Computing*, Volume 1, Issue 1.

BIAC/ICC (2003*), Information Security Assurance for Executives*, BIAC/ICC, Paris.

BIAC/ICC (2004), *Securing your Business: Information Security Issues and Resources for Small and Entrepreneurial Companies*, BIAC/ICC, Paris.

Bitzer, J., W. Schrettl and P.J.H. Schroeder (2007), "Intrinsic Motivation in Open Source Software Development", *Journal of Comparative Economics*.

Braunschweig, B. (2005), "Software Interoperability for Petroleum Applications", *Oil & Gas Science and Technology* – Rev. IFP, Vol. 60 (2005), No. 4.

Brewer, J. (2004), "Web Accessibility Highlights and Trends", Proceedings of the 2004 International Cross-disciplinary Workshop on Web Accessibility (W4A), referenced at *http://portal.acm.org/citation.cfm?id=990657&dl=GUIDE&coll=GUIDE*.

BSA (2007), BSA Statement on Interoperability: Innovation, Choice, and the Role of Governments, Business Software Alliance, Brussels.

CNSS (2006), National Information Assurance Glossary, CNSS Instruction No. 4009, Committee on National Security Systems, US National Security Agency, Ft. Meade, Maryland, June, www.cnss.gov/Assets/pdf/cnssi_4009.pdf.

Cusumano, M. (2004), *The Business of Software*, Free Press, New York.

Dewar, R. and E. Schonberg (2008), "Computer Science Education: Where Are the Software Engineers of Tomorrow?", *Crosstalk*, *www.stsc.hill.af.mil/CrossTalk/2008/01/0801DewarSchonberg.html*.

Donzelli, P., F. Shull, S. Asgari and V. Basili (2006) "Evolving a Dependability Requirements Elicitation and Modeling Framework Based on Use", Technical Report CS-TR-4851, University of Maryland, November, *www.lib.umd.edu/drum*.

EICTA (2004), "Interoperability", EICTA White Paper, Belgium.

Gasser, U. and J. Palfrey, (2007), "Breaking Down Digital Barriers: When and How ICT Interoperability Drives Innovation", Berkman Publication Series, available at *http://cyber.law.harvard.edu/interop/downloads.html*.

Gulliksen, J., and S. Harker (2004), "The Software Accessibility of Human-Computer Interfaces – ISO Technical Specification 16071", Universal Access in the Information Society, Vol. 3.

IDC (2007), "DC Says Expanding Needs for Mobile Device Security Software Will Drive Revenue Growth over the Next Five Years", press release, *www.idc.com/getdoc.jsp?containerId=prUS20633707*.

IEEE (2004), Guide to the Software Engineering Body of Knowledge (SWEBOK), A. Abran and J.W. Moore (executive eds.) and P. Bourque and R. Dupuis (eds.), Los Alamitos, CA, USA. (SWEBOK is also published as ISO/IEC TR 19759:2005.)

IFIP (1988), "The Charter of WG 10.4 on Dependable Computing And Fault Tolerance" (established 1980, revised 1988), International Federation for Information Processing, *www.dependability.org*.

Kavcic, A. (2005), "Software Accessibility: Recommendation and Guidelines", EUROCON 2005, *http://lgm.fri.uni-lj.si/alenka/web/Eurocon2005.pdf*.

Lueders, H. (2005), "Interoperability and Open Standards for eGovernment Services", Initiative for Software Choice (ISC), Brussels.

Lyu, M.R. (2007), "Software Reliability Engineering: A Roadmap", International Conference on Software Engineering, 2007 Future of Software Engineering.

MacWilliams, A. (2004), "Software Development Challenges for Ubiquitous Augmented Reality", Technical University of Munich, Germany.

Niemelä, E. and J. Latvakoski (2004), "Survey of Requirements and Solutions for Ubiquitous Software", Mobile Ubiquitous Computing Conference 2004, Washington, DC.

Nikkei (2003), "Informatization State Survey", *Nikkei Computer* (magazine in Japanese), as cited by the Ministry of Economy, Trade and Industry, *www.nikkeibp.com/html/pub/computer.html*.

OECD (1980), *OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*, OECD, Paris.

OECD (2002), *Guidelines for the Security of Information Systems and Networks: Towards a Culture of Security*, OECD, Paris.

OECD (2004a), "Summary of Responses to the Survey on the Implementation of the *OECD Guidelines for the Security of Information Systems and Networks: Towards a Culture of Security*", OECD, Paris.

OECD (2004b), "Digital Broadband Content: Mobile Content, New Content for New Platforms", OECD, Paris.

OECD (2005), "The Promotion of a Culture of Security in OECD Countries », OECD, Paris.

OECD (2006), "Mobile Commerce", OECD, Paris.

OECD (2008a), "Shaping Policies for the Future of the Internet Economy", OECD, Paris, available at *www.oecd.org/futureinternet*.

OECD (2008b), *Computer Viruses and Other Malicious Software (Malware): A Threat to the Internet Economy*, OECD, Paris.

Pan, Jiantao (1999), "Software Reliability", Carnegie Mellon University, available at: *www.ece.cmu.edu/~koopman/des_s99/sw_reliability/#reference*.

Parker, G. and M.W. Van Alstyne (2008), "Innovation, Openness, and Platform Control", 24 July, *http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1079712*.

Reed, P.S., D. Gardner-Bonneau and S. Isensee (2004), "Software Accessibility Standards and Guidelines: Progress, Current Status, and Future Development", *Universal Access in the Information Society, Vol. 3*.

Sousa, J.P. and D. Garlan (2002), "Auro: an Architectural Framework for User Mobility in Ubiquitous Computing Environments", 3rd Working IEEE/IFIP Conference on Software Architecture, Montreal, Canada.

TEITAC (2008), Report to the Access Board: Refreshed Accessibility Standards and Guidelines in Telecommunications and Electronic and Information Technology, April, *www.access-board.gov/sec508/refresh/report/#1*.

Tsilas, N.L. (2007), "Enabling Open Innovation and Interoperability: Recommendations for Policy-Makers", ACM International Conference Proceeding Series, Vol. 232, 1st International Conference on Theory and Practice of Electronic Governance, Macao, China.

Voas, J., and Wilbanks, L. (2008), "Information and Quality Assurance: An Unsolved, Perpetual Problem for Past and Future Generations", *IT Pro*, May/June.

Weiler, J.H.H. (2005), "The Legal Regulation of Software Interoperability in the EU", Jean Monnet Working Paper 07/05, New York.

Yoshida, M. (2006), Japan Perspectives on ICT Accessibility, presentation at CSUN 21st Annual International Conference.

Annex A

Review of Studies on Market Penetration Rates

This annex outlines several studies on market penetration rates. The results are presented here in order to illustrate attempts to estimate the size of the software market taking into account software that is not fully covered in the general statistics. In particular, many of these studies focus on open source software (OSS).[1] *This review is presented in three sections:* 1) *survey-based studies,* 2) *census-like studies, and* 3) *country-focused studies.*

General survey-based studies

Saugatuck Technology (2007) indicates that open source now accounts for roughly 10% of worldwide key enterprise on-premises software. The analysis projects that the OSS may expand its share to between 15 and 20% by 2010. The firm also expects explosive growth in mixed-source environments and that more than 35% of new commercial software implementations will include open source components by that year. The authors foresee that four or five master brands (some coming from a traditional software background) will influence or control at least 30% of the open source marketplace by 2012. Regarding other market forecasts, Gartner (2008) predicts that by 2012, more than 90% of enterprises will use open source in direct or embedded forms,

Concerning the market for operating systems, Gartner (2004) presented a study based on sales figures from computer vendors with pre-installed Linux. This study concluded that the Linux operating system was on 5% of personal computers sold worldwide in 2004. This, however, was not confirmed by the census – like studies by Net Applications (presented below).

Regarding office applications, there are a variety of sources referring to surveys and to the number of downloads published on the Open Office website. According to Jupiter Research (2003) Open Office had a 6% market share among small and medium enterprises in 2003. For 2007, Freeform Dynamics (2007) estimated that market share of Open Office amounts to 7% for office use and 20% for home use. With respect to studies of a more narrow geographic scope, according to Méndez (2005) Open Office has 8.5% of the market among major North American companies, whereas in Germany according to TechConsult (2006) it is likely to be used by 8% of firms.

Census-like surveys

The best known of these market share indicators is the monthly Web Server Survey conducted by the research company Netcraft *(http://news.netcraft.com/)*. It provides a "census" of accessible web servers on the Internet of which over a half run the OSS Apache web server application. However, in recent years Apache suffered a significant decline in share, which is related to the growing market share of Microsoft's products and the introduction of Google's new product, GFE (Google Front End).

Another automated census-like survey is provided by Timme (2004); this presents market penetration rates for mail servers. Timme performed a scan on existing mail servers in Europe and in the US based on IP addresses. The most popular programs are listed in Table A.1. OSS servers accounted for almost half of the total market.

Table A.1. Share of web servers

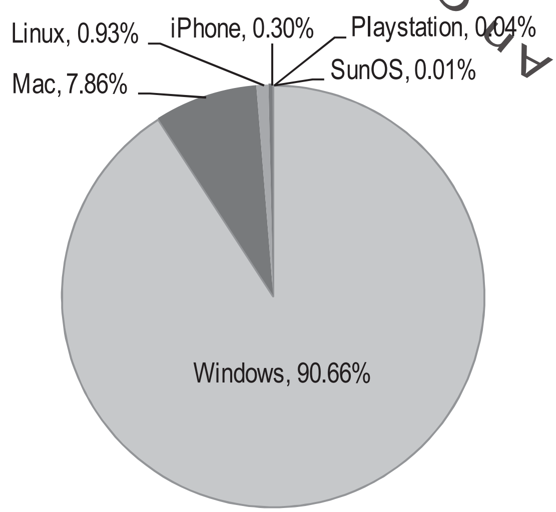
Mail server	Market share (%)
Microsoft ESMTP MAIL Service	23.89
Sendmail*	22.38
IMail	19.25
Exim*	13.22
Postfix*	5.57
MailEnable	0.84
Merak	0.29
Mailer Daemon	0.24
MERCUR	0.13
CommuniGate Pro	0.12
Xmail	0.10
Lotus Domino	0.07
Microsoft Exchange	0.07
NTMail	0.06
DynFX	0.03
Kerio MailServer	0.03
GroupWise	0.02
InterScan	0.02
CMailServer	0.00
Netscape Messaging Server	0.00
Qmail*	0.00
WinRoute Pro	0.00
Unknown/other	13.66

*OSS software.

Source: Timme (April 2004), mail server survey, *www.falkotimme.com*.

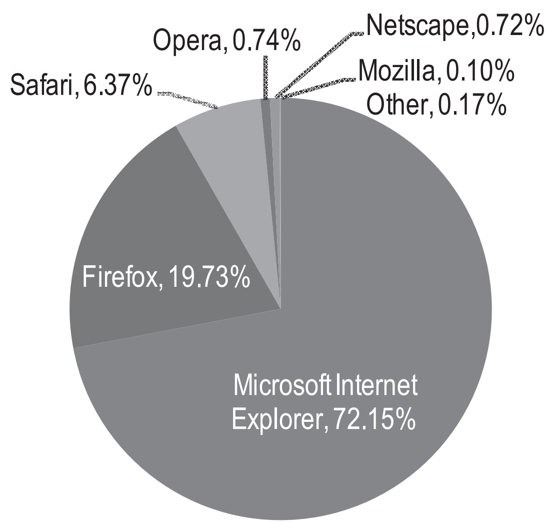
Finally, Net Applications[2] provides data on market shares of web browsers and operating systems based on census-based surveys. The results presented by Net Applications come from a scan of the visitors browsers to the "exclusive on demand network of small to medium enterprise live stats customers". According to the results of Net Application's scan, MS Windows has a dominant position on the operating systems' market (Figure A.1) and MS Explorer is the most popular web browser (Figure A.2).

Figure A.1. Share of operating systems, October 2007



Source: www.NetApplications.com.

Figure A.2. Share of Web browsers, August 2008



Source: www.NetApplications.com.

Country-focused studies

Concerning the geographical scope of available analyses, the most broad and advanced studies consider the European Union, the United States, Japan and Brazil.

European Union

The use of OSS in the European Union has advanced in part in relation to actions by public authorities. Certain local or regional European public administrations were among the early adopters of OSS and were already running OSS-based systems and applications in the late 1990s. Today, the European Commission services run a multitude of OSS-based applications. They initiate and sponsor new OSS projects for internal and external use. The rationale and benefits behind the strategy are to promote: broader choice of solutions; reduced captivity towards any particular product; faster and cheaper delivery of IT services by customisation and deployment of OSS solutions as opposed to full-scale development; and improvement of the IT staff skills through hands-on experience with emerging technologies.[3] Open source related services are expected to reach 32% share of all IT services by 2010, and open source related activities might account for as much as 4% of European GDP by 2010 (EC, 2006).

Three major survey-based studies present recent OSS penetration rates for the European Union.

Méndez (2005) analysed the adoption of OSS in the European industry. He found that European firms have been recently increasing their adoption rates of OSS. Méndez estimated that at the end of 2005, 40% of EU firms were using open source solutions and another 8% had declared plans to introduce such systems. The telecommunications industry, mass-media and public sector entities were the most intense users of OSS, with 45% using OSS for key applications.

IDC (2005) presents the Western European Software End-User Survey of 625 firms. According to the results, over 40% of respondents reported "significant, some or limited" use of OSS for operating systems purposes and nearly 60% declared use of open source databases.

The third study on the OSS penetration rate in the EU was done by MERIT (2006). According to the results presented, the OSS market share is higher in Europe than in the US for some types of software (such as operating systems), which in turn is followed by Asia. These market shares have seen considerable growth between 2000 and 2005.

A survey run by MERIT in Germany, UK and Sweden differentiated responses by type of software and firm size. According to the results, the highest OSS penetration rates were found in Germany where 30.7% of small and 30.6% of large firms[4] were found to use OSS operating systems. The smallest penetration rates were noted in UK large firms, where only 2% used OSS on desktops and 3.7% used OSS server operating systems (see Table A.2).

Table A.2. OSS usage by type of software in Germany, Sweden and UK

Percentages

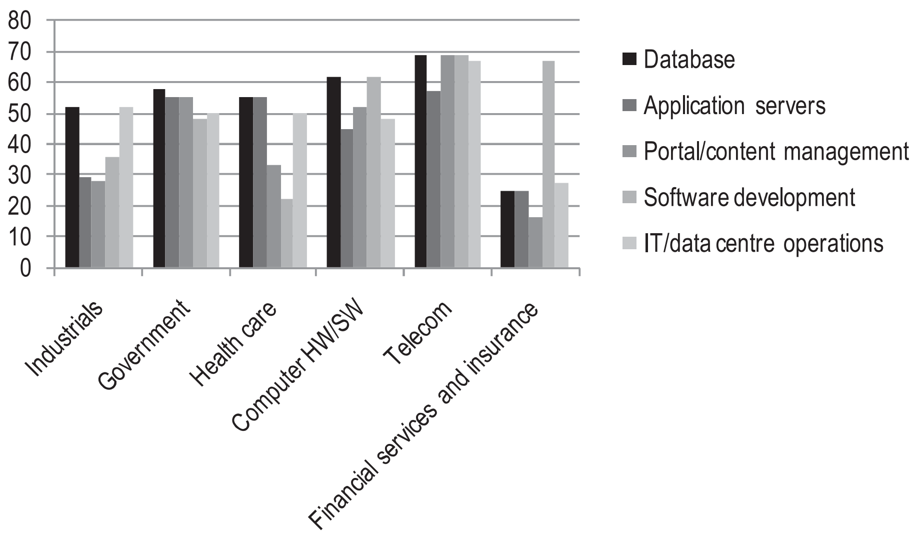
	Germany		Sweden		United Kingdom		Total
	Small	Large	Small	Large	Small	Large	
OSS as server for operating system	30.7	30.6	9.8	11.0	8.1	3.7	15.7
OSS for databases	14.1	20.8	7.5	8.2	13.3	4.6	11.1
OSS on the desktop	13.7	6.5	3.4	3.2	7.6	2.0	6.9
OSS for websites	15.8	17.3	7.5	8.7	7.9	4.3	10.1

Source: MERIT (2006), *Economic Impact of FLOSS on Innovation and Competitiveness of the EU ICT Sector*, UNU-MERIT, Maastricht University.

United States

Walli *et al*. (2005) in a study on OSS penetration rates in the United States conclude that a clear majority of US companies and government institutions are adopting OSS for some uses. Their survey results indicated that 87% of the 512 interviewed firms were using OSS programs. The usage of OSS goes beyond operating systems (*i.e.* Linux) to include various applications (Figure A.3).

Figure A.3. OSS usage by industry and type of software in the United States



Source: Walli *et al.* (2005), *The Growth of Open Source Software in Organizations*, Optaros Publications and Thought Leadership, Boston.

Walli *et al*. (2005) find that the bigger companies (larger than USD 50 million) are more likely to implement OSS solutions, with the telecommunications business being the leader in terms of OSS adoption intensity. The financial services industry reports the lowest OSS penetration rate. However, this is the industry with the highest number of declarations of planned introduction of open source software solutions.

Japan

There are no official data regarding OSS in Japan, although the OSS market is said to be expanding. The lack of data is mainly due to the difficulty of economic measurement for products that have a non-commercial character. However, reports from private firms provide some insights. For example, NEC Corporation, a strong promoter of Linux in Japan, estimates that the total size of the Linux-related business continues to expand by approximately 20% a year and will be JPY 575 billion in 2009 (Table A.3).

Table A.3. Estimated Linux-related business in Japan

JPY 100 million[5]

	2005	2006	2007	2008	2009
Linux system integration	1 200	1 600	2 000	2 300	2 600
Support	900	1 000	1 100	1 300	1 650
Linux platform (SW, HW)	700	900	1 100	1 300	1 500
Total	2 700	3 500	4 200	4 900	5 750

Source: NEC Corporation.

The most active adopter of OSS in Japan is the public sector, especially municipal governments. According to a survey on the introduction of OSS/Linux, the ratio for public sector entities with at least one OSS/Linux server is over 90% in 2005 (Table A.4). Another survey covering local governments shows that municipals recognise the necessity of OSS literacy; over nine out of ten persons in charge of ICT at municipal offices want to participate in the OSS community.

Table A.4. OSS/Linux server introduction ratio, by sector

Entities that have introduced at least one OSS/Linux server, percentages

	2004 (n = 637)	2005 (n = 793)
Public/public interest corporations	61.9	93.1
Financial/insurance industry	22.0	47.5
Service	44.8	60.6
Distribution	23.2	40.1
Assembly manufacturing	29.0	43.8
Process manufacturing	23.3	35.6
Construction	26.9	54.1
Total	32.0	48.8

Source: Yano Research Institute Ltd.

There are several municipal administrations that have decided to migrate systems to OSS; this approach is based on comparison studies concluding that OSS systems would be an effective way to reduce costs in their long-term operations. Urazoe-city in Okinawa, for example, started to introduce Linux in 2002, and has already shifted about 500 desktop PCs to Linux-based systems. The remaining PCs – a further 500 – will be done in a couple of years.

Contrary to municipalities, small and medium-size firms are sluggish in their OSS adoption. A survey of small and medium-size firms, of which annual turnover is between five hundred million and JPY 5 billion, counts no more than 6.7% as the share of Linux powered servers (OS) in 2007, as compared to a combined share of over 90% for Windows NT, Windows 2000 and Windows 2003 (Table A.5). More than 60% of the firms state that they do not have the intention to adopt Linux as a server OS. This figure increased by 3.1 percentage points from the previous year's survey. Over 50% of the firms declining to adopt Linux cite as a reason the limited number of engineers in their companies who have Linux literacy; around 40% say they are satisfied with Windows.

Table A.5. Server operating system software, share of usage in small and medium size firms

Percentages

	2006 (n = 1 876)	**2007** (n = 2 487)
Windows NT	18.8	14.5
Windows 2000	39.2	38.5
Windows 2003	25.9	35.8
Linux	5.5	6.7
UNIX	3.3	1.7
Netware	0.5	0.6
OS/2	0.5	0.1
Other	6.3	2.0

Source: Nork Research Co., Ltd., (2007).

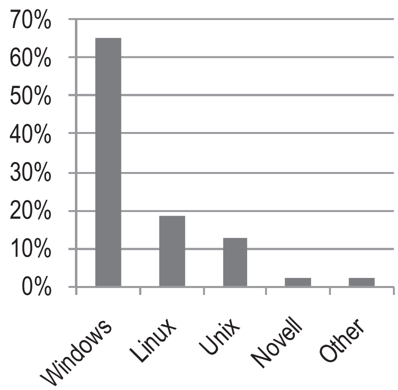
In the realm where software is used as a component in other products such as embedded software, OSS has begun to play an important role. All of the Japanese makers of non-CRT television sets, liquid-crystal display TVs and plasma display TVs have adopted Linux as a platform for embedded software. A similar transition is on the move with mobile phones. In January 2007, NEC Corporation, Panasonic Mobile Communications and NTT Docomo set up a foundation to promote Linux usage as a mobile phone platform, and now co-operate with Motorola, Samsung and Vodafone on this.

Brazil

Softex (2005) presents a comprehensive report on the role and impact of open source software in the Brazilian economy. The study outlines results of a few surveys on OSS penetration rates in selected fragments of the software market in Brazil. The report concludes that OSS is a growing phenomenon in Brazil, particularly within the Brazilian industry. The study points to the Linux operating system as the trigger for further adoption of other OSS solutions in a single firm.

Two other studies present statistics on OSS penetration rates in Brazil. The first one is the study by Fortes (2004), analysing 100 of the "most connected" firms according to *Info Exame* magazine. According to his results 64% of these companies already use the Linux operating system. A newer study by Meirelles (2007) adds to the picture by showing some general statistics for the Brazilian economy as a whole. According to Meirelles' results most of the Brazilian industry relies on non-OSS operating systems (Figure A.5).

Figure A.5. Operating systems, market penetration rates in Brazilian firms, 2006 and 2007



Source: Meirelles, (2007), Tecnologia de Informação (Information Technology), São Paulo: EAESP/FGV, *www.eaesp.fgvsp.br*.

A similar situation to the market for operating systems was found by Meirelles (2007) for spreadsheets used by Brazilian firms, where a large majority uses a proprietary program (Table A1.6).

Table A1.6. Spreadsheets used by Brazilian firms, 2006 and 2007

Percentages

	Penetration rate
Excel	92
Open and Star Office	6
Other	2

Source: Meirelles, (2007), Tecnologia de Informação (Information Technology), São Paulo: EAESP/FGV, *www.eaesp.fgvsp.br*.

Notes

1. The measurement OSS is difficult by nature, given that the reuse and redistribution can occur without the author's authorisation and given that a portion of OSS distribution occurs without any monetary payment involved.
2. Available at *www.netapplications.com*.
3. This information was based on a document obtained at: *www.ec.europa.eu/idabc/en/document/7389/5998* (last accessed in 2007). Current information is available from the Open Source Observatory and Repository for European public administrations at *http://osor.eu/* (last accessed on 27 September 2008).
4. The threshold value between a small and a large firm in MERIT's survey was 500 employees.
5. As of 4 October 2007, JPY 100 million amounted to approximately USD 861 395 (Source: *www.oanda.com*).

References

EC (2006), Study on the Economic Impact of Open Source Software on Innovation and the Competitiveness of the Information and Communication Technologies (ICT) Sector in the EU, Rishab Aiyer Ghosh, European Commission, MERIT, Contract ENTR/04/112.

Fortes, D.O. (2004), "O fenómeno do software livre" (The free software phenomenon), *Revista Info*, 19 (215).

Freeform Dynamics (2007), "Office Suite Wars, Microsoft vs. OSS vs. SaaS", available at *www.freeformdynamics.com*.

Gartner (2004), "Linux Has a Fight on Its Hands in Emerging PC Markets", Gartner Inc.

Gartner (2008), "The State of Open Source, 2008", Gartner, Inc.

IDC (2005) "European End-User Survey: 2005 Spending Priorities, Outsourcing, Open Source, and Impact of Compliance", available at *www.idc.com*.

Jupiter Research (2003), "Spending $2 Billion on 45 Million: Microsoft's Sales Assault Targets SMBs", Jupiter Research, July.

Meirelles, F.S. (2007), *Tecnologia de Informação* (Information Technology), EAESP/FGV, São Paulo, *www.eaesp.fgvsp.br*.

Méndez, M.Á. (2005), "Is Open Source Gaining Adoption In Europe?", Forrester Research, available at *www.forrester.com*.

MERIT (2006), Economic Impact of FLOSS on Innovation and Competitiveness of the EU ICT Sector, UNU-MERIT, Maastricht University.

Saugatuck Technology (2007), "Open Source: The Next Disruptive IT Influence", *www.saugatech.com/395order.htm*.

Softex (2005), "Impact of the Free Software and Open Source on the Software Industry in Brazil", available at *www.softex.br*.

TechConsult (2006), "Der deutsche Markt für Linux und Open Source (The German Marketfor Linux and Open Source)" TechConsult GmbH, available at *www.techconsult.de*.

Timme, F. (2004), Mail Server Survey: April 2004, available at *www.falkotimme.com*.

Walli, S., D. Gynn and B. von Rotz (2005), "The Growth of Open Source Software in Organizations", Optaros Publications and Thought Leadership, Boston.

Annex B

Beyond the Traditional Software Sector: The Role of Businesses as Software Developers and Users[1]

While much of the buzz about software in the popular press is focused on consumers, businesses beyond the traditional software sector are playing a very significant role as both users and developers of software. Firms across the economy are deploying large amounts of software to power their operations and products. As discussed in Chapters 1 and 2, the amount of software in many types of products is increasing rapidly. For example, the number of lines of source code in a mobile phone is expected to increase from 2 million today to 20 million by 2010; by that time, a car may contain 100 million lines of code (Charette, 2005; Ito, 2007). The growth of software involves far more than a technical evolution. It will have consequences for companies, industries and national economies.

One indication of the extent of the interrelationship between the software sector and other industries can be found in the composition of employment by sector. Table B.1 presents data for the European Union on the share of computer specialists in total employment of various sectors, as calculated drawing on European labour force survey data. As might be expected, the top-ranked industries are "computer and related activities" and "manufacture of office equipment and computers". Perhaps more notable is the engagement of significant numbers of computer specialists (which includes software professionals) across a wide range of sectors, from collection, purification and distribution of water to post and telecommunications. Among the NACE 2-digit sectors shown in the table, all have 1.4% or more of their employment in the occupational classification of computer specialists. A similar measure is presented for the United States in Table B.2, though at a higher level of sectoral aggregation. Here again, the presence of computer professionals, including software professionals, is notable across the economy. Only four of the 13 sectors presented have less than 1% of total employment in the computer professional category. Business and professional services account for the largest share of both computer-related and software-related employment, followed by the information sector, financial activities and the public sector.

Implications at company level

The growth in the volume of software needed to deliver the functionality demanded in products across the economy poses challenges to producers beyond the traditional software sector. To respond using in-house development approaches, typically entails significant growth of the software engineering and support resources. Companies must assess whether to attempt such development keeping in mind the cost of building such capacity and the challenge of finding the necessary qualified staff (*e.g.* software engineers).[2]

Table B.1. Top 25 ranking of industries according to their share of computer-related employment in total employment, EU15, 2006

EU15 - NACE Sector	%
72 Computer and related activities	53.5
30 Manufacture of office machinery and computers	23.4
32 Manufacture of radio, television and communication equipment and apparatus	6.8
73 Research and development	5.2
66 Insurance and pension funding, except compulsory social security	5.1
64 Post and telecommunications	4.7
65 Financial intermediation, except insurance and pension funding	4.5
33 Manufacture of medical, precision and optical instruments, watches and clocks	3.7
99 Extra-territorial organizations and bodies	3.1
40 Electricity, gas, steam and hot water supply	3.1
67 Activities auxiliary to financial intermediation	3.0
11 Extraction of crude petroleum and natural gas; service activities incidental to oil and gas extraction excluding surveying	2.7
35 Manufacture of other transport equipment	2.3
62 Air transport	2.1
31 Manufacture of electrical machinery and apparatus n.e.c.	2.0
24 Manufacture of chemicals and chemical products	1.8
22 Publishing, printing and reproduction of recorded media	1.8
74 Other business activities	1.7
34 Manufacture of motor vehicles, trailers and semi-trailers	1.6
41 Collection, purification and distribution of water	1.6
21 Manufacture of pulp, paper and paper products	1.6
71 Renting of machinery and equipment without operator and of personal and household goods	1.5
75 Public administration and defence; compulsory social security	1.5
29 Manufacture of machinery and equipment n.e.c.	1.4
23 Manufacture of coke, refined petroleum products and nuclear fuel	1.4

Note: The ISCO categories "213 computing professionals" and "312 computer associate professionals" were used as a proxy.

Source: Authors' calculations based on the European Labour Force Survey.

Table B.2. The share of computer-related employment and software and programming specific employment in total employment, United States, 2004

Percentages

Industry	Computer-related	Software + programming
1 Agriculture, forestry, fishing and hunting	0.2	0.2
2 Mining	1.9	1.2
3 Construction	0.1	0.0
4 Manufacturing	2.6	1.2
5 Wholesale and retail trade	1.3	0.5
6 Transportation and utilities	1.4	0.5
7 Information	8.8	3.4
8 Financial activities	3.8	1.6
9 Professional and business services	9.3	4.0
10 Educational and health services	1.1	0.2
11 Leisure and hospitality	0.4	0.1
12 Other services	0.6	0.2
13 Public administration	3.6	1.2
Total	2.4	0.9

Note: The CPS categories "110 computer and information systems managers", "1000 computer scientists and systems analysts", "1010 computer programmers", "1020 computer software engineers", "1040 computer support specialists", "1060 database administrators", "1100 network and computer systems administrators", and "1110 network systems and data communications analysts" are used. The ISCO categories "213 computing professionals" and "312 computer associate professionals" are used as a proxy.

Source: Authors' calculations based on US Current Population Survey (CPS).

Box B.1. Embedded Software: Illustration of the Successful Launching of An Affiliated Producer

Starting an affiliated software business will change the economics of a parent firm that formerly produced embedded software in-house, potentially in a beneficial way.

In shifting the embedded software activity to a new affiliate, additional costs will be incurred to establish the marketing and sales capacity for the software formerly sold with the hardware, but now brought independently to the market. On top of that, the legal infrastructure has to be built to protect the software and the associated intellectual property.

How might the economics work in a successful case? Let us assume that half of the costs in building a software business are related to developing and maintaining the software and the other half are related to the business aspects of software, such as marketing, sales and legal expenses. For the parent firm, bringing the software to the market via a separate entity means doubling the costs in this case. On the other hand, if the software affiliate is then successful in selling the software to its mother company as well as to, say, two other companies then it can begin to reap some offsetting economies of scale. Consider the comparison in the following table.

Economics of software production in-house versus an affiliate

In-house production		Shift to sourcing via an affiliated software vendor	
Sales (marketed as part of the hardware products)	0	Sales:	
		To parent company	75
		To two other companies (for same price)	150
Cost: software development	100	Costs:	
		Development:	100
		Software marketing and sales:	100
Result	-100		25

In this example of a successful case, the parent company obtains the software it needs for 25% less than if it had developed the software internally and its affiliate earned a profit of 25. By increasing the volume of production, the unit costs were reduced and the parent firm benefitted directly and via the affiliate. In this way, software started to create value for the company involved.

If the software is to be developed in-house for an embedded application, then the projects are generally financed from existing operations. Some companies have margins or market shares that allow them to bear the increasing software costs without excessive problems. Examples can be found in specific high-volume consumer electronics products (where economies of scale can be reaped) or some high margin medical devices. For such firms, there may be strategic advantages or relatively lower costs in taking the in-house development approach. However, in other cases, the software development burden would be excessive for a single producer. The computer industry faced this over recent decades. It may be on the horizon for individual car manufacturers, who may lack in-house capacity for development of the many large or specialised applications that are foreseen (*e.g.* guidance or safety systems). Thus, the choice of how to proceed with software development is challenging an increasing number of manufacturers and service-sector firms on an on-going basis.

This choice is not always an easy one, particularly because management perceptions of investment in software development may vary. In cases where software is not generating a separate revenue stream, it may be viewed by management as an operating cost – one that may be growing, and something to be minimised. On the other hand, the entry of new suppliers of such software on the market for inputs or the entry of competitors to the manufacturer on the consumer market for the product may signal that a given sector is becoming more dynamic. Management may then perceive a strategic or tactical advantage in engaging more deeply in the software development process including via increased openness and collaboration with partners beyond the walls of the firm or even launching a new software venture.

Implications for industries

The growth of software in terms of size, complexity and range of application poses challenges to existing firms. In cases where firms can consolidate fragmented demand across markets and supply more standardised software solutions, there is the potential for them to introduce disruptive and beneficial change. The computer industry was completely changed in the 1980s and 1990s as a result of opening of systems to permit independent software supply. As an article in *The Wall Street Journal* pointed out (WSJ, 2005), “Until the 1980s, a handful of giant manufacturers controlled the design, construction and sale of their machines, along with most of the software that ran on them. But then came along the personal computer, which relied on standardised chips and software.” There are indications that similar revolutions may be set to happen in other industries.

Volume is a key element in meeting the growing software demands. The digital, intangible nature of software implies low marginal costs. This can enable software suppliers to have better returns overall and per unit in cases when volume is large. Moreover, the larger software companies have accumulated significant software R&D capabilities that can be put to use in other industries that are struggling to meet the need for software. In view of such factors, this section considers three illustrative industry case studies where software may prove transformative in coming years: automobiles, mobile phones and healthcare.

Software growth in the automobile industry

Software is playing an increasingly important role in delivery of the functionalities sought by automotive customers, with the result that the size of programs is growing enormously. If projections are accurate, then the software bundle for a fully-equipped car may grow from about 1 million lines of code in 2000 to some 100 million lines of code (about twice the size of Microsoft’s Vista operating system) by 2010. At the same time, individual car producers control just a fraction of the market and will have trouble to mobilise the sales volume needed to justify the investment in development of such massive amounts of code in house.[3]

As a result, it is not surprising that auto manufacturers are already reaching out to buy-in the necessary functionality. In some cases, this is happening indirectly through external purchases of hardware units with embedded software such as multimedia devices and navigation systems. These are, however, rather isolated pieces of hardware. One response can be seen in the Japan Automotive Software Platform Initiative (Jaspar), which is aiming to pool resources from across the Japanese auto industry to address automotive software challenges in a manner that yields economies of scale.[4] From outside of the sector, major software suppliers have targeted the automotive sector as a market for development. For example, Microsoft's *Windows Automotive* is an open platform aiming to facilitate third party developer participation in creation of automotive software solutions.[5]

Liability issues remain to be resolved and may impede opening and development of some aspects of automotive software. Given that much of the software in a car relates to safety and critical operational features, it is more closely integrated than, for example, a multimedia system. This may impede the opening of some embedded systems to independent outside software suppliers.

Software growth in the mobile phone industry

The number of mobile phones sold in 2008 in the world is estimated to be about 1.3 billion. The size of the software bundle in these products is growing rapidly and is slated to increase from an estimated 2 million lines in 2006 to 20 million in 2010. The volume leaders in the mobile phone industry are the large technology firms such as Nokia, Samsung and Motorola.[6] Nokia alone produces over 500 million mobile phones a year.

The amount of software in high-end phones or smartphones is increasing especially rapidly. (A smartphone is typically defined as a mobile phone with an open operating system.) Moving to smartphones can be seen as a way in which the mobile phone industry tries to handle the increasing amounts of software required to deliver the desired functionality in their products. By providing a more open operating system, it allows multiple software companies to supply the ever-increasing amounts of software required providing a competitive phone. Dominant operating systems in mobile phones today are Symbian and Windows Mobile. It brings the mobile phone industry from the embedded stage into the open system stage.

Two recent events indicate that a portion of the market for some types of software in the mobile phone industry may shifting towards open source approaches. First, in October 2007, Google announced its platform Android, which it claims will be the first complete, open and free mobile software platform (OHA, 2008). Then, in June 2008, Nokia announced its purchase of the Symbian operating system with the intention to bring the system into the open source domain. Many of the key players in the mobile phone industry have engaged in the Android or Symbian initiatives.

One accelerator for the changes in the mobile phone market may be the fact that only five manufacturers supply the vast majority of mobile phones. They have clear volume leadership and may be in a position to agree upon standards that facilitate software development. Moreover, the expanding technological capacity of the phones is making it possible to add new functionalities; nowadays, a vast amount of software required to produce a competitive mobile phone.

Software growth in the healthcare industry

The aging population around the world is providing impetus for improved software functionality. As with cars and mobile phones, the amount of software in healthcare electronics products is growing rapidly especially for medical diagnostics equipment and information management systems (storage and exchange) in medical institutions. ITEA[7] (2005) estimates that R&D expenditure for software in the medical equipment industry will grow from EUR 7 billion in 2002 to EUR 28 billion in 2015. Software R&D is slated to increase its overall share of R&D in the sector from 25% to 33%. Moreover, ITEA estimates that software staff may account for up to 60% of the R&D staff for major providers.

In relation to information management, one of the drivers for software demand in the sector is the expansion of work on human DNA. New insights concerning DNA are enabling more personalised healthcare (*e.g.* different therapies for different people with the same disease), but also requiring availability of detailed information about a patient's DNA and healthcare history.

In the past, many medical systems employed closed embedded systems, but the growth in demand for software is leading equipment manufacturers to consider ways to collaborate and leverage their efforts. One factor that may impede this development is, again, concern about liability issues. Similarly to the automobile industry, some medical equipment suppliers may opt to keep control of critical aspects of the software in order to contain the risk. The role of regulatory bodies may also impede the growth of small software companies in this business, in view of sometimes lengthy and expensive approval procedures.

Summing up

The evolution of software is impacting industries across the economy and appears set to transform sectors such as the car, mobile phone and healthcare industries. Software is becoming increasingly mainstreamed in the products and services of modern life. In many instances, the markets for such software remain fragmented with many firms still developing software internally or opening their systems only partially to outside collaboration or suppliers. Liability concerns may play a role in this regard. On the other hand, there are tremendous possibilities for economies of scale in software and firms are striving to find ways to consolidate the growing demands for software and thereby reap the benefits.

Notes

1. This section draws on an issues paper prepared for the OECD by M. van Genuchten (2008), as well as other sources.

2. There is strong demand across the economy for qualified software professionals, and some manufacturers – where software may be viewed as a sideline – may be at a competitive disadvantage (*e.g.* in terms of image as perceived by software professionals) *vis-à-vis* leading software or computer service firms in terms of recruitment.

3. Toyota and General Motors each sold approximately 9.37 million cars in 2007 (BBC news, 2008), in an overall market of approximately 60 million cars.

4. As stated on the Jaspar website, "JasPar will strive to reduce technology development costs and promote technology development by encouraging Japanese companies to collaboratively develop pre-competitive technologies such as automotive LAN enabling technology, middleware and software platform", *www.jaspar.jp/english/guide/purpose.php*, (last accessed on 7 August 2008).

5. A fact sheet on Window's Automotive is available at: *http://download.microsoft.com/download/f/b/5/fb5efead-ef87-4ddc-a05b-2d75154e0edc/WA_Datasheet.pdf* (last accessed on 7 August 2008).

6. In the second quarter of 2008, the respective market shares as posted on Imran's Everything Cellular website are: Nokia (41.1%), Samsung (15.4%) and Motorola (9.5%), available at: *www.mobileisgood.com/statistics.php#current* (last accessed on 7 August 2008).

7. ITEA stands for Information Technology for European Advancement. It is "a strategic pan-European programme for advanced pre-competitive R&D in software for Software-intensive Systems and Services". For more information see: *www.itea-office.org* (last accessed on 7 August 2008).

References

Charrette, R. N. (2005), "Why Software Fails" IEEE Spectrum, September.

ITEA (2005), "Software Intensive Systems in the Future", Information Technology for European Advancement, *www.itea-office.org/publications*.

Ito, S. (2007), "Developing Global Society through Innovation in the Software Sector", presentation at the OECD Conference on Innovation in The Software Sector, Caceres, Spain, 30 November.

OHA (2008), Open Handset Alliance, *www.openhandsetalliance.com*.

Van Genuchten, M. (2008), "The Impact of Software Growth on Companies, Industries and National Economies", issues paper prepared for the meeting of the OECD Advisory Expert Group on Innovation in the Software Sector, Tokyo, 7 October.

WSJ (2005), "Shakeout Could Jolt Electronics", *Wall Street Journal*, 27 January.

OECD PUBLISHING, 2, rue André-Pascal, 75775 PARIS CEDEX 16
PRINTED IN FRANCE
(93 2009 05 1 P) ISBN 978-92-64-06361-7 – No. 57025 2009

Innovation in the Software Sector

This book puts a spotlight on innovation across the software universe, setting out key issues and highlighting policy perspectives. It spans research and development, invention, production, distribution and use of software in the market. It also covers core innovation themes from a user perspective – including security and privacy, mobility, interoperability, accessibility and reliability.

For more information about this book and other OECD work in this area, see ***www.oecd.org/sti/innovation/software***.

Innovation in the Software Sector is part of the OECD Innovation Strategy, a comprehensive policy strategy to harness innovation for stronger and more sustainable growth and development, and to address the key global challenges of the 21st century. For more information about the OECD Innovation Strategy, see ***www.oecd.org/innovation/strategy***.

The full text of this book is available on line via this link:
www.sourceoecd.org/scienceIT/9789264063617

Those with access to all OECD books on line should use this link:
www.sourceoecd.org/9789264063617

SourceOECD is the OECD's online library of books, periodicals and statistical databases.
For more information about this award-winning service and free trials, ask your librarian, or write to us at **SourceOECD@oecd.org**.

OECD*publishing*
www.oecd.org/publishing

ISBN 978-92-64-06361-7
93 2009 05 1 P

9 789264 063617